2-24-2020

# Advancing Multi-Modal Deep Learning: Towards Language-Grounded Visual Understanding

Kushal Kafle
kk6055@rit.edu

# ADVANCING MULTI-MODAL DEEP LEARNING: TOWARDS LANGUAGE-GROUNDED VISUAL UNDERSTANDING

by

Kushal Kafle

B.E. in Electronics and Communication Engineering,
Institute of Engineering, Tribhuvan University, 2012

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

Feb 24, 2020

Signature of the Author _____

Accepted by _____
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

---

Ph.D. DEGREE DISSERTATION

---

The Ph.D. Degree Dissertation of Kushal Kafle has been examined
and approved by the dissertation commitee as satisfactory for the dissertation
required for the Ph.D. degree in Imaging Science

Dr. Christopher Kanan, Dissertation Advisor

Dr. Matt Huenerfauth, External Chair

Dr. Nathan D. Cahill

Dr. Raymond Ptucha

Dr. Chenliang Xu

Date

*To Jwala, the love of my life*

# ADVANCING MULTI-MODAL DEEP LEARNING: TOWARDS LANGUAGE-GROUNDED VISUAL UNDERSTANDING

by

Kushal Kafle

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

## Abstract

Using deep learning, computer vision now rivals people at object recognition and detection, opening doors to tackle new challenges in image understanding. Among these challenges, understanding and reasoning about language grounded visual content is of fundamental importance to advancing artificial intelligence. Recently, multiple datasets and algorithms have been created as proxy tasks towards this goal, with visual question answering (VQA) being the most widely studied. In VQA, an algorithm needs to produce an answer to a natural language question about an image. However, our survey of datasets and algorithms for VQA uncovered several sources of dataset bias and sub-optimal evaluation metrics that allowed algorithms to perform well by merely exploiting superficial statistical patterns. In this dissertation, we describe new algorithms and datasets that address these issues. We developed two new datasets and evaluation metrics that enable a more accurate measurement of abilities of a VQA model, and also expand VQA to include new abilities, such as reading text, handling out-of-vocabulary words, and understanding data-visualization. We also created new algorithms for VQA that have helped advance the state-of-the-art for VQA, including an algorithm that surpasses humans on two different chart question answering datasets about bar-charts, line-graphs and pie charts. Finally, we provide a holistic overview of several yet-unsolved challenges in not only VQA but vision and language research at large. Despite enormous progress, we find that a robust understanding and integration of vision and language is still an elusive goal, and much of the progress may be misleading due to dataset bias, superficial correlations and flaws in standard evaluation metrics. We carefully study and categorize these issues for several vision and language tasks and outline several possible paths towards development of safe, robust and trustworthy AI for language-grounded visual understanding.

## Acknowledgements

First and foremost, I would like to thank my advisor Dr. Christopher Kanan. He has provided me a perfect mix of guidance during my early years, and freedom during my later years. His guidance and mentoring has not only been instrumental in shaping this dissertation but also shaped my scientific outlook and undoubtedly will shape my scientific endeavors for years to come. I am honored to have had the privilege of working with him for the past five years.

I would also like to thank everyone on my dissertation committee, Dr. Matt Huenerfauth, Dr. Nathan Cahill, Dr. Raymod Ptucha and Dr. Chenliang Xu for providing valuable suggestions and critiques about my work. I would also like to thank my mentors from my two very valuable internship experiences, Dr. Scott Cohen and Dr. Brian Price from Adobe Research and Dr. Dinei Florencio from Microsoft.

Many thanks to the Chester F. Carlson department of Imaging Science for providing an excellent environment for learning, research and collaboration. I am especially thankful to Dr. David Messinger for his support and all the administrative members, Sue, Mel, and Beth, that provided valuable support allowed me to focus on my research.

I am grateful and honored to have worked with incredible lab-mates. I am especially thankful to Ron for his valuable discussions, tips and proof-readings of many of my works during my early years. I am thankful to all my brilliant co-authors from my lab, Robik, Manoj and Tyler, for their valuable discussions, support and patience. Thanks to Ryne for incredibly valuable insights and agreeing to review many of my papers on a short notice.

Next, I would like to thank all the friends that made life outside the lab more enjoyable, especially Emily, Ryan, Colin and Jared. You have been a source of endless support and have made an enormous difference in my well-being and made my day-to-day Ph.D. life enjoyable.

I would also like to thank my parents for instilling a sense of curiosity from an early age. I would also like to thank both my parents and my parents-in-law, for their support and encouragement to pursue a Ph.D.

Finally, I would like to thank my wife, Jwala Dhamala, who has been a constant source of love and support for the last 10 years of my life. She has always believed in me, sometimes more than I believed in myself, and has been my biggest source of strength and perseverance. Despite pursuing her own Ph.D., she was able to support not only my physical and emotional well-being but also support my research via proof-reading, suggestions, re-touching my diagrams/tables and valuable discussions.

# Author Publications

† indicates that a modified version of this publication is included in this proposal.

**Refereed Publications**

† **Kafle, K.** and Kanan, C. (2016) Answer-type prediction for visual question answering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

† **Kafle, K.** and Kanan, C. (2017) Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding (CVIU)*, 163: 3-20.

- **Kafle, K.**, Yousefhussein, M., and Kanan, C.. (2017) Data augmentation for visual question answering. *International Natural Language Generation Conference (INLG)*.

† **Kafle, K.** and Kanan, C. (2017) An analysis of visual question answering algorithms. *International Conference on Computer Vision (ICCV)*.

† **Kafle, K.**, Cohen, S., Price, B., and Kanan, C. (2018). DVQA: Understanding Data Visualizations via Question Answering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Acharya, M., **Kafle, K.** and Kanan, C. (2019) TallyQA: Answering Complex Counting Questions *Association for the Advancement of Artificial Intelligence (AAAI)*.

- Shrestha, R., **Kafle, K.** and Kanan, C. (2019) Answer Them All! Toward Universal Visual Question Answering Models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

† **Kafle, K.**, Shrestha, R., and Kanan, C. (2019) Challenges and Prospects in Vision and Language Research. *Frontiers in Artificial Intelligence*, 2: 28.

† **Kafle, K.**, Shrestha, R., Price, B., Cohen, S., and Kanan, C. (2020). Answering Questions about Data Visualizations using Efficient Bimodal Fusion. *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction and Motivation

As humans, we can recognize entities, process their attributes, and understand the relationships and interactions between different entities in our visual field. One of the grand goals in computer vision is to develop algorithms that allow machines to perceive and process the visual data on par with or beyond human capabilities.

Progress has been swift in several *narrow* computer vision tasks, such as image classification [3, 4], object detection [5, 6], and activity recognition [7–9]. Most of the progresses in these problems can be attributed to the efficacy of deep convolutional neural networks (CNNs), which, when paired with sufficiently large annotated dataset, can even rival human capacities in certain well-defined tasks [4]. Consequently, there is a strong desire in the computer vision community to push the frontiers and pursue grander challenges. One of the such grand problems is to enable machines to integrate and process both natural language and linguistic concepts.

Numerous works have been explored on combining vision with language, including image and video captioning [10, 11], visual question answering (VQA) [12–17], referring expression recognition (RER) [18], image retrieval [19, 20], activity recognition [21, 22], and language-guided image generation [23, 24]. We collectively refer to these tasks as vision and language (V&L) tasks. Among all V&L tasks, VQA is one of the most-studied problem. VQA requires an algorithm to answer arbitrary text-based questions about images [12, 25]. A robust VQA system must be capable of not only solving a wide range of classical computer vision and NLP tasks but also an successful integration and grounding of visual content to language input. We place special emphasis on VQA in this dissertation.

A wide variety of algorithms have been proposed for each of the aforementioned tasks, producing increasingly better results across datasets. However, several studies have called into question the *true* capability of these systems and the efficacy of current assessment methods [17, 26, 27]. Systems are heavily influenced by dataset bias and lack robustness to uncommon visual configurations [15, 17, 27], but these are often not measured and call

into question the value of these benchmarks. These issues also impact system assessment and deployment. Systems can amplify spurious correlations between gender and potentially unrelated variables in V&L problems [22, 28], resulting in the possibility of severe negative real-world impact.

In this dissertation, we explore the design and development of datasets, algorithms, and evaluation of language grounded visual understanding problems in presence of strong sources of biases that impact both training and evaluation. We place particular emphasis on VQA but we also outline how similar issues exist for a wider variety of V&L tasks.

## 1.1 Objectives

There are three major objectives in this dissertation:

1. Critically evaluate the efficacy of datasets, evaluation metrics and algorithms for language grounded visual understanding
   (a) Determine the extent of biases in VQA datasets and how they affect the performance evaluation (Chapter 2 and 4)
   (b) Evaluate how different design choices for VQA algorithms affect their performance. (Chapter 2)
   (c) Assess the performance of VQA algorithms for out-of-vocabulary words in questions and answers (Chapter 5)
   (d) Study the effects of bias and other challenges in existing vision and language (V&L) tasks beyond VQA (Chapter 7)
2. Develop novel tasks and evaluation metrics that address the shortcomings in existing vision and language problems
   (a) Study efficacy of task-directed categorization of visual questions in mitigating the biases in VQA (Chapter 4)
   (b) Quantify the effects of long-tailed distribution, class distribution, and the presence of nonsensical queries on VQA algorithms (Chapter 2)
   (c) Develop a dataset with a simpler and more carefully controlled visual properties to allow bias-free test for elementary visual reasoning (Chapter 5)
3. Develop novel algorithms for language grounded visual understanding that work robustly for various real-world usages
   (a) Assess whether predicting the *type* of answer to a question in VQA can help in answering the question (Chapter 3)

(b) Assess whether algorithms developed for natural-image VQA translate to chart question answering (CQA) (Chapter 5)

(c) Enable VQA systems capable of optical character recognition (OCR) and parsing out-of-vocabulary questions and answers (Chapter 5)

(d) Demonstrate efficacy of better bi-modal fusion techniques for chart question answering (Chapter 6)

## 1.2 Dissertation Layout

This dissertation consists of eight chapters with a modified version of current introductory chapter serving as Chapter 1 and the conclusion as Chapter 8. Chapters 2–7 contain materials from papers that we published in various refereed venues. A brief overview of contents of each chapter is presented below.

### 1.2.1 Chapter 2: Datasets, Algorithms, and Challenges in VQA

VQA is a recent problem in computer vision and natural language processing that has garnered a large amount of interest from the deep learning, computer vision, and natural language processing communities. Since 2014, many datasets and algorithms have been released. In this chapter, we critically examine the current state of VQA in terms of problem formulation, existing datasets, evaluation metrics, and algorithms. In specific, we describe various datasets and preset their limitations with regards to their ability to properly train and assess VQA algorithms. We then exhaustively review existing VQA algorithms. Finally, we discuss possible future directions in VQA research, some of which we explore in later chapters.

This chapter contains modified and updated materials from our CVIU paper entitled "Visual Question Answering: Datasets, Algorithms and Future Challenges" [13] and our Frontiers paper entitled "Challenges and Prospects in Vision and Language Research" [29].

### 1.2.2 Chapter 3: Answer-Type Prediction for VQA

In this chapter, we describe a novel approach capable of answering open-ended text-based questions about images, which is known as Visual Question Answering (VQA). The key intuition of our approach is that we can predict the form of the answer from the question before providing the final answer. We formulate this in a Bayesian framework and combine with a discriminative model. This algorithm was among the earliest developed for VQA, and it achieved state-of-the-art results on four benchmark datasets for open-ended VQA: DAQUAR, COCO-QA, VQA 1.0, and Visual7W.

This chapter contains modified contents from our CVPR-2016 paper entitled "Answer-Type Prediction for Visual Question Answering" [30].

### 1.2.3 Chapter 4: Analysis of VQA Algorithms

Popular VQA datasets have critical flaws in both their content and the way algorithms are evaluated on them. As a result, evaluation scores are inflated and often predominantly determined by an algorithm's ability to answer easy questions. All of this makes it difficult to compare different methods. In this chapter, we analyze existing VQA algorithms using a new dataset called the Task Driven Image Understanding Challenge (TDIUC), which has over 1.6 million questions organized into 12 different categories (available for download at `https://goo.gl/Ng9ix4`). TDIUC introduces questions that are meaningless for a given image to force a VQA system to reason about image content. We propose new evaluation schemes that compensate for over-represented question-types and make it easier to study the strengths and weaknesses of algorithms. We analyze the performance of both baseline and state-of-the-art VQA models, including multi-modal compact bilinear pooling (MCB), neural module networks, and recurrent answering units. Our experiments establish how attention helps certain categories more than others, determine which models work better than others, and explain how simple models (e.g., MLP) can surpass more complex models (MCB) by simply learning to answer large, easy question categories.

This chapter contains a modified version of our ICCV-2017 paper entitled "An Analysis of Visual Question Answering Algorithms" [17].

### 1.2.4 Chapter 5: Understanding Data Visualizations via Question Answering

In this chapter, we explore VQA systems for non-image data. Specifically, we describe data and models required for automatically parsing and understanding information contained in bar-charts. Bar charts are an effective way to convey numeric information, but today's algorithms cannot parse them. Existing methods fail when faced with even minor variations in appearance. Here, we present DVQA, a dataset that tests many aspects of bar chart understanding in a question answering framework. Unlike visual question answering (VQA), DVQA requires processing words and answers that are unique to a particular bar chart. State-of-the-art VQA algorithms perform poorly on DVQA, and we propose two strong baselines that perform considerably better. The work described in this chapter will enable algorithms to automatically extract numeric and semantic information from vast quantities of bar charts found in scientific publications, Internet articles, business reports, and many other areas. Furthermore, the DVQA dataset described in this chapter tests several important technical and practical problems not tackled by existing VQA algorithms.

This chapter contains a modified version of our CVPR-2018 paper entitled "DVQA: Understanding Data Visualizations via Question Answering" [31].

### 1.2.5 Chapter 6: Answering Questions about Data Visualizations using Efficient Bimodal Fusion

In this chapter, we describe a novel algorithm for chart question answering that greatly surpasses existing baselines for two recently introduced datasets; DVQA, which we described in Chapter 5 and FigureQA [2]. Our CQA algorithm is called parallel recurrent fusion of image and language (PReFIL). PReFIL first learns bimodal embeddings by fusing question and image features and then intelligently aggregates these learned embeddings to answer the given question. Despite its simplicity, PReFIL greatly surpasses state-of-the art systems for both DVQA and FigureQA dataset and even surpasses the human-level accuracy on the FigureQA dataset. Since we did not originally collect human baselines for DVQA, we also collect crowd-sourced answers to a portion of DVQA questions to estimate human baselines. Under identical conditions, PReFIL surpasses existing baseline algorithm on DVQA by over 40% and also surpasses human accuracy estimates when an oracle OCR is used. Additionally, we demonstrate that PReFIL can be used to reconstruct tables by asking a series of questions about a chart.

This chapter contains a modified version of our WACV-2020 paper entitled "Answering Questions about Data Visualizations using Efficient Bimodal Fusion" [32].

### 1.2.6 Chapter 7: Challenges in Vision and Language Research

In this chapter, we endeavor to outline several challenges that are still unresolved in the vision and language research, including VQA. Ideally, V&L tasks should test a plethora of capabilities that integrate computer vision, reasoning, and natural language understanding. However, rather than behaving as visual Turing tests, we argue that the state-of-the-art systems could be achieving good performance through flaws in datasets and evaluation procedures. While earlier chapters in this dissertation have focused mostly on VQA, this chapter aims to highlight challenges in a broader set of V&L problems. In chapter 2, we discussed several issues in VQA and we proposed a solution in chapter 4 to mitigate certain forms of dataset bias. We also discussed other attempted solutions to the dataset and language bias issues in VQA in the literature. However, many challenges still remain and affect a large number of V&L tasks, including VQA. In this chapter, we document several such issues by drawing on several recent studies, including our own.

This chapter contains a materials from our Frontiers paper entitled "Challenges and Prospects in Vision and Language Research" [29].

### 1.2.7 Chapter 8: Conclusion and Future Works

In this chapter, we briefly review the progress made by this dissertation towards the grand goal of language-guided visual learning. Then, we will highlight several concrete directions for . We will focus on two key research directions. Firstly, we discuss the future of chart-question answering, where we have made considerable advancements in the available datasets. We detail how we can now tackle bigger challenges than are posed by existing CQA datasets. Secondly, we focus on the future of V&L research in a broader sense. In chapter 7, we outlined several key issues in V&L research that could give a false sense of progress. We discuss future research directions to mitigate several of these issues which we hope will pave a path towards development of more robust and capable V&L models in the future.

This chapter contains a materials from our Frontiers paper entitled "Challenges and Prospects in Vision and Language Research" [29] and our WACV-2020 paper entitled "Answering Questions about Data Visualizations using Efficient Bimodal Fusion" [32].

# Chapter 2

# Datasets, Algorithms, and Challenges in Visual Question Answering

## 2.1 Introduction

Advancements in deep learning and the availability of large-scale datasets have resulted in great progress in computer vision and natural language processing (NLP). Deep convolutional neural networks (CNNs) have enabled unprecedented improvements in classical computer vision tasks, e.g., image classification [33] and object detection [34]. Similarly, various deep learning based approaches have enabled enormous advances in classical NLP tasks, e.g., named entity recognition [35], sentiment analysis [36], question-answering [37,38] and dialog systems [39]. With annotated datasets rapidly increasing in size thanks to crowd-sourcing, similar outcomes can be anticipated for other focused computer vision problems. However, these problems are narrow in scope and do not require holistic understanding of images. Building upon these advances in computer vision and NLP, there is a push to attack new problems that enable concept comprehension and reasoning capabilities to be studied at the intersection of vision and language (V&L) understanding. There are numerous applications for V&L systems, including enabling the visually impaired to interact with visual content using language, human-computer interaction, and visual search. Human-robot collaboration would be greatly enhanced by giving robots understanding of human language to better understand the visual world. As humans, we can identify the objects in an image, understand the spatial positions of these objects, infer their attributes and relationships to each other, and also reason about the purpose of each object given the surrounding context. We can ask arbitrary questions about images and also communicate the information gleaned from them.

Until recently, developing a computer vision system that can answer arbitrary natural language questions about images has been thought to be an ambitious, but intractable, goal. However, since 2014, there has been enormous progress in developing systems

with these abilities. Visual Question Answering (VQA) is a computer vision task where a system is given a text-based question about an image, and it must infer the answer. Questions can be arbitrary and they encompass many sub-problems in computer vision, e.g.,

- Object recognition - What is in the image?
- Object detection - Are there any cats in the image?
- Attribute classification - What color is the cat?
- Scene classification - Is it sunny?
- Counting - How many cats are in the image?

Beyond these, there are many more complex questions that can be asked, such as questions about the spatial relationships among objects (What is between the cat and the sofa?) and common sense reasoning questions (Why is the the girl crying?). A robust VQA system must be capable of solving a wide range of classical computer vision tasks as well as needing the ability to reason about images.

There are many potential applications for VQA. The most immediate is as an aid to blind and visually impaired individuals, enabling them to get information about images both on the web and in the real world. For example, as a blind user scrolls through their social media feed, a captioning system can describe the image and then the user could use VQA to query the image to get more insight about the scene. More generally, VQA could be used to improve human-computer interaction as a natural way to query visual content. A VQA system can also be used for image retrieval, without using image meta-data or tags. For example, to find all images taken in a rainy setting, we can simply ask 'Is it raining?' to all images in the dataset. Beyond applications, VQA is an important basic research problem. Because a good VQA system must be able to solve many computer vision problems, it can be considered a component of a Turing Test for image understanding [40, 41].

A Visual Turing Test rigorously evaluates a computer vision system to assess whether it is capable of human-level semantic analysis of images [40, 41]. Passing this test requires a system to be capable of many different visual tasks. VQA can be considered a kind of Visual Turing Test that also requires the ability to understand questions, but not necessarily more sophisticated natural language processing. If an algorithm performs as well as or better than humans on arbitrary questions about images, then arguably much of computer vision would be solved. But, this is only true if the benchmarks and evaluation tools are sufficient to make such bold claims.

In this review, we discuss existing datasets and methods for VQA. We place particular emphasis on exploring whether current VQA benchmarks are suitable for evaluating whether a system is capable of robust image understanding. In Section 2.2, we compare VQA with other computer vision tasks, some of which also require the integration of vision and language (e.g., image captioning). Then, in Section 2.3, we describe currently

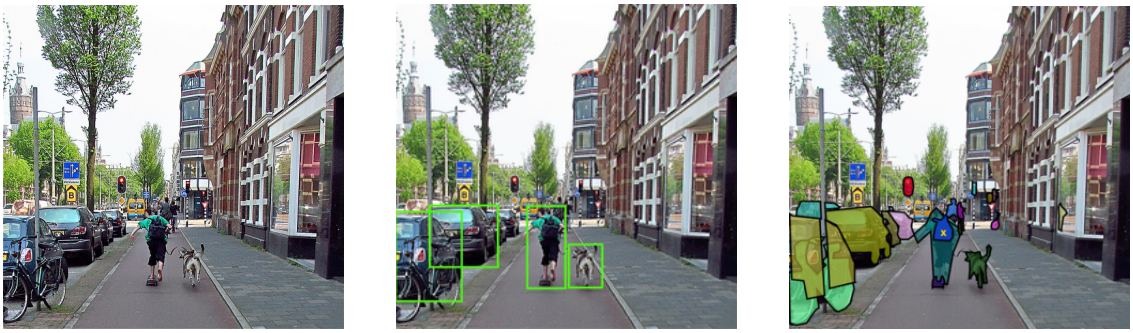


Figure 2.1: Object detection, semantic segmentation, and image captioning compared to VQA. The middle figure shows the ideal output of a typical object detection system, and the right figure shows the semantic segmentation map from the COCO dataset [34]. Both tasks lack the ability to provide contextual information about the objects. The captions for this COCO image range from very generic descriptions of the scene, e.g., *A busy town sidewalk next to street parking and intersections.*, to very focused discussion of a single activity without qualifying the overall scene, e.g., *A woman jogging with a dog on a leash.* Both are acceptable captions, but significantly more information can be extracted with VQA. For the COCO-VQA dataset, the questions asked about this image are *What kind of shoes is the skater wearing?*, *Urban or suburban?*, and *What animal is there?*

available datasets for VQA with an emphasis on their strengths and weaknesses. We discuss how biases in some of these datasets severely limit their ability to assess algorithms. In Section 2.4, we discuss the evaluation metrics used for VQA. Then, we review existing algorithms for VQA and analyze their efficacy in Section 2.5. Finally, we discuss possible future developments in VQA and open questions.

## 2.2 Vision and Language Tasks Related to VQA

### 2.2.1 Elementary Visual Understanding Vs. VQA

The overarching goal of VQA is to extract question-relevant semantic information from the images, which ranges from the detection of minute details to the inference of abstract scene attributes for the whole image, based on the question. While many computer vision problems involve extracting information from the images, they are limited in scope and generality compared to VQA. Object recognition, activity recognition, and scene classification can all be posed as image classification tasks, with today's best methods doing this using CNNs trained to classify images into particular semantic categories. The most successful of these is object recognition, where algorithms now rival humans in accuracy [4].

But, object recognition requires only classifying the dominant object in an image without knowledge of its spatial position or its role within the larger scene. Object detection involves the localization of specific semantic concepts (e.g., cars or people) by placing a bounding box around each instance of the object in an image. The best object detection methods all use deep CNNs [5,6,42]. Semantic segmentation takes the task of localization a step further by classifying each pixel as belonging to a particular semantic class [43,44]. Instance segmentation further builds upon localization by differentiating between separate instances of the same semantic class [45–47].

While semantic and instance segmentation are important computer vision problems that generalize object detection and recognition, they are not sufficient for holistic scene understanding. One of the major problems they face is label ambiguity. For example, in Figure 2.1, the assigned semantic label for the position of the yellow cross can be 'bag', 'black,' or 'person.' The label depends on the task. Moreover, these approaches alone have no understanding of the role of an object within a larger context. In this example, labeling a pixel as 'bag' does not inform us about whether it is being carried by the person, and labeling a pixel as 'person' does not tell us if the person is sitting, running, or skateboarding. This is in contrast with VQA, where a system is required to answer arbitrary questions about images, which may require reasoning about the relationships of objects with each other and the overall scene. The appropriate label is specified by the question.

### 2.2.2 Image Captioning Vs. VQA

One of the most studied and closest problem to VQA is image captioning [7, 10, 48–50], in which an algorithm's goal is to produce a natural language description of a given image. Image captioning is a very broad task that potentially involves describing complex attributes and object relationships to provide a detailed description of an image.

However, there are several problems with the visual captioning task, with evaluation of captions being a particular challenge. The ideal method is evaluation by human judges, but this is slow and expensive. For this reason, multiple automatic evaluation schemes have been proposed. The most widely used caption evaluation schemes are BLEU [51], ROUGE [52], METEOR [53], and CIDEr [54]. With the exception of CIDEr, which was developed specifically for scoring image descriptions, all caption evaluation metrics were originally developed for machine translation evaluation. Each of these metrics has limitations. BLEU, the most widely used metric, is known to have the same score for large variations in sentence structure with largely varying semantic content [55]. For captions generated in [56], BLEU scores ranked machine generated captions above human captions. However, when human judges were used to judge the same captions, only 23.3% of the judges ranked the captions to be of equal or better quality than human captions. While other evaluation metrics, especially CIDEr and METEOR, show more robustness

in terms of agreement with human judges, they still often rank automatically generated captions higher than human captions [57].

One reason why evaluating captions is challenging is that a given image can have many valid captions, with some being very specific and others generic in nature (see Figure 2.1). However, captioning systems that produce generic captions that only superficially describe an image's content are often ranked high by the evaluation metrics. Generic captions such as 'A person is walking down a street' or 'Several cars are parked on the side of the road' that can be applicable to a large number of images are often ranked highly by evaluation schemes and human judges. In fact, a simple system that returns the caption of the training image with the most similar visual features using nearest neighbor yields relatively high scores using automatic evaluation metrics [58].

Dense image captioning (DenseCap) avoids the generic caption problem by annotating an image densely with short visual descriptions pertaining to small, but salient, image regions [59]. For example, a DenseCap system may output 'a man wearing black shirt,' 'large green trees,' and 'roof of a building,' with each description accompanied by a bounding box. A system may generate a large number of these descriptions for rich scenes. Although many of these descriptions are short, it is still difficult to automatically assess their quality. DenseCap can also omit important relationships between the objects in the scene by only producing isolated descriptions for each regions. Captioning and DenseCap are also task agnostic and a system is not required to perform exhaustive image understanding.

In conclusion, a captioning system is at liberty to arbitrarily choose the level of granularity of its image analysis which is in contrast to VQA, where the level of granularity is specified by the nature of the question asked. For example, 'What season is this?' will require understanding the entire scene, but 'What is the color of dog standing behind the girl with white dress?' would require attention to specific details of the scene. Moreover, many kinds of questions have specific and unambiguous answers, making VQA more amenable to automated evaluation metric than captioning. Ambiguity may still exist for some question types (see Section 2.4), but for many questions the answer produced by a VQA algorithm can be evaluated with one-to-one matching with the ground truth answer.

### 2.2.3 Other Vision and Language Tasks

Besides VQA, there is a significant amount of recent work that combines vision with language. Bidirectional sentence-to-image and image-to-sentence retrieval problems are among the earliest V&L tasks [19]. Early works dealt with simpler keyword-based image retrieval [19], with later approaches using deep learning and graph-based representations [20]. Visual semantic role labeling requires recognizing activities and semantic context in images [21, 22]. Image captioning, described earlier, is the task of generating descriptions for visual content, involves both visual and language understanding. It

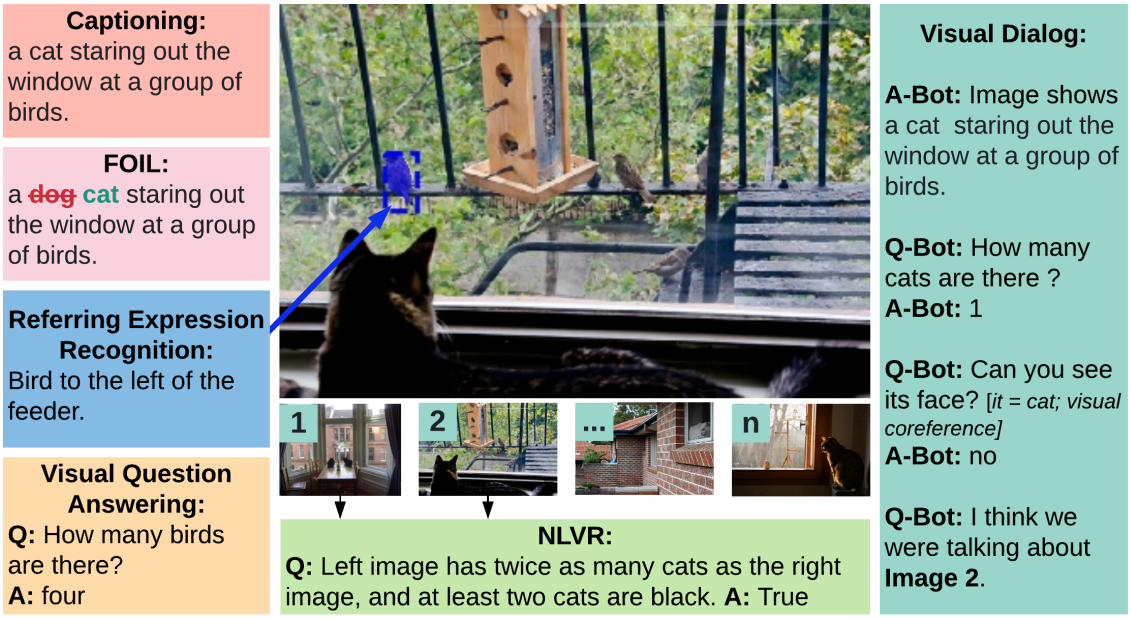



Figure 2.2: Common tasks in vision and language research.

requires describing the gist of the *interesting* content in a scene [34, 60], while also capturing specific image regions [59]. Video captioning adds the additional complexity of understanding temporal relations [11].

Another related task, called referring expression recognition (RER) requires models to provide visual evidence by either selecting among a list of possible image regions or generating bounding boxes that correspond to input phrases [18, 61, 62]. Since the output of an RER query is *always* a single box, it is often quite easy to *guess* the correct box. To counter this, [63] proposed visual query detection (VQD), a form of goal-directed object detection, where the query can have 0–15 valid boxes making the task more difficult and more applicable to real-world applications. FOIL takes a different approach and requires a system to differentiate invalid image descriptions from valid ones [64]. Natural Language Visual Reasoning (NLVR) requires verifying if image descriptions are true [65, 66].

In another closely related task, EmbodiedQA requires the agent to explore its environment to answer questions [67]. The agent must actively perceive and reason about its visual environment to determine its next actions. In visual dialog, an algorithm must hold a conversation about an image [68, 69]. In contrast to VQA, visual dialog requires understanding the conversation history, which may contain visual co-references that a system must resolve correctly. The idea of conversational visual reasoning has also been explored in Co-Draw [70], a task where a *teller* describes visual scenes and a *drawer* draws them without looking at the original scenes.

## 2.3 Datasets for VQA

Beginning in 2014, five major datasets for VQA have been publicly released. These datasets enable VQA systems to be trained and evaluated. As of this article, the main datasets for VQA are DAQUAR [25], COCO-QA [71], The VQA Dataset [12], FM-IQA [72], Visual7W [73], and Visual Genome [74]. With exception of DAQUAR, all of the datasets include images from the Microsoft Common Objects in Context (COCO) dataset [34], which consists of 328,000 images, 91 common object categories with over 2 million labeled instances, and an average of 5 captions per image. Visual Genome and Visual7W use images from Flickr100M in addition to the COCO images. A portion of The VQA Dataset contains synthetic cartoon imagery, which we will refer to as SYNTH-VQA. Consistent with other papers [30, 75, 76], the rest of The VQA Dataset will be referred as COCO-VQA, since it contains images from the COCO image dataset. Table 2.1 contains statistics for each of these datasets.

An ideal VQA dataset needs to be sufficiently large to capture the variability within questions, images, and concepts that occur in real world scenarios. It should also have a fair evaluation scheme that is difficult to 'game' and doing well on it indicates that an algorithm can answer a large variety of question types about images that have definitive answers. If a dataset contains easily exploitable biases in the distribution of the questions or answers, it may be possible for an algorithm to perform well on the dataset without really solving the VQA problem.

In the following subsections, we critically review the available datasets. We describe how the datasets were created and discuss their limitations.

### 2.3.1 DAQUAR

The DAtaset for QUestion Answering on Real-world images (DAQUAR) [25] was the first major VQA dataset to be released. It is one of the smallest VQA datasets. It consists of 6795 training and 5673 testing QA pairs based on images from the NYU-DepthV2 Dataset [77]. The dataset is also available in an even smaller configuration consisting of only 37 object categories, known as DAQUAR-37. DAQUAR-37 consists of only 3825 training QA pairs and 297 testing QA pairs. In [78], additional ground truth answers were collected for DAQUAR to create an alternative evaluation metric. This variant of DAQUAR is called DAQUAR-consensus, named after the evaluation metric. While DAQUAR was a pioneering dataset for VQA, it is too small to successfully train and evaluate more complex models. Apart from the small size, DAQUAR contains exclusively indoor scenes, which constrains the variety of questions available. The images tend to have significant clutter and in some cases extreme lighting conditions (see Figure 2.3). This makes many questions difficult to answer, and even humans are only able to achieve 50.2% accuracy on the full dataset.

Table 2.1: Statistics for VQA datasets using either open-ended (OE) or multiple-choice (MC) evaluation schemes.

| | DAQUAR [25] | COCO-QA [71] | COCO-VQA [12] | FM-IQA [72] [1] | Visual7W [73] | Visual genome [74] |
|---|---|---|---|---|---|---|
| Total Images | 1,449 | 123,287 | 204,721 | 120,360 | 47,300 | 108,000 |
| QA Pairs | 12,468 | 117,684 | 614,163 | 250,569 | 327,939 | 1,773,258 |
| Distinct Answers | 968 | 430 | 105,969 | N/A | 65,161 | 207,675 |
| % covered by top-1000 | 100 | 100 | 82.8 | N/A | 56.29 | 60.8 |
| % covered by top-10 | 25.04 | 19.71 | 51.13 | N/A | 17.13 | 13.07 |
| Human Accuracy | 50.2 | N/A | 83.3 | N/A | 96.6 | N/A |
| Longest Question | 25 words | 24 words | 32 words | N/A | 24 words | 26 words |
| Longest Answer | 7 (list of 1 words) | 1 word | 17 words | N/A | 20 words | 24 words |
| Avg. Answer Length | 1.2 words | 1.0 words | 1.1 words | N/A | 2.0 words | 1.8 words |
| Image Source | NYUDv2 | COCO | COCO | COCO | COCO | COCO, YFCC |
| Annotation | Manual+Auto | Auto | Manual | Manual | Manual | Manual |
| Evaluation Type | OE | OE | MC or OE | OE | MC or OE | OE |
| Question Types | 3 | 4 | - | - | - | - |


COCO-QA: What does an intersection show on one side and two double-decker buses and a third vehicle,?
Ground Truth: Building

DAQUAR: What is behind the computer in the corner of the table?
Ground Truth: papers

Figure 2.3: Sample images from DAQUAR and the COCO-QA datasets and the corresponding QA pairs. A significant number of COCO-QA questions have grammatical errors and are nonsensical, whereas DAQUAR images are often marred with clutter and low resolution images.

### 2.3.2 COCO-QA

In COCO-QA [71], QA pairs are created for images using an Natural Language Processing (NLP) algorithm that derives them from the COCO image captions. For example, using the image caption `A boy is playing Frisbee`, it is possible to create the question `What is the boy playing?` with `frisbee` as the answer. COCO-QA contains 78,736 training and 38,948 testing QA pairs. Most questions ask about the object in the image (69.84%), with the other questions being about color (16.59%), counting (7.47%) and location (6.10%). All of the questions have a single word answer, and there are only 435 unique answers. These constraints on the answers makes evaluation relatively straightforward.

The biggest shortcoming of COCO-QA is due to flaws in the NLP algorithm that was used to generate the QA pairs. Longer sentences are broken into smaller chunks for ease of processing, but in many of these cases the algorithm does not cope well with the presence of clauses and grammatical variations in sentence formation. This results in awkwardly phrased questions, with many containing grammatical errors, and others being completely unintelligible (see Figure 2.3). The other major shortcoming is that it only has four kinds of questions, and these are limited to the kinds of things described in COCO's captions.

### 2.3.3 The VQA Dataset

The VQA Dataset [12] consists of both real images from COCO and abstract cartoon images. Most work on this dataset has focused solely on the portion containing real world imagery from COCO, which we refer to as COCO-VQA. We refer to the synthetic portion of the dataset as SYNTH-VQA.

COCO-VQA consists of three questions per image, with ten answers per question. Amazon Mechanical Turk (AMT) workers were employed to generate questions for each image by being asked to 'Stump a smart robot,' and a separate pool of workers were hired to generate the answers to the questions. Compared to other VQA datasets, COCO-VQA consists of a relatively large number of questions (614,163 total, with 248,349 for training, 121,512 for validation, and 244,302 for testing). Each of the questions is then answered by 10 independent annotators. The multiple answers per question are used in the consensus-based evaluation metric for the dataset, which is discussed in Section 2.4.

SYNTH-VQA consists of 50,000 synthetic scenes that depict cartoon images in different simulated scenarios. Scenes are made from over 100 different objects, 30 different animal models, and 20 human cartoon models. The human models are the same as those used in [79], and they contain deformable limbs and eight different facial expressions. The models also span different age, gender, and races to provide variation in appearance. SYNTH-VQA has 150,000 QA pairs with 3 questions per scene and 10 ground truth answers per question. By using synthetic images, it becomes possible to create a more varied and balanced dataset. Natural image datasets tend to have more consistent context and biases, e.g., a street scene is more likely to have picture of a dog than a zebra. Using synthetic images, these biases can be reduced. Yin and Yang [14] is a dataset built on top of SYNTH-VQA that tried to eliminate biases in the answers people have to questions. We further discuss Yin and Yang in Section 2.6.1.

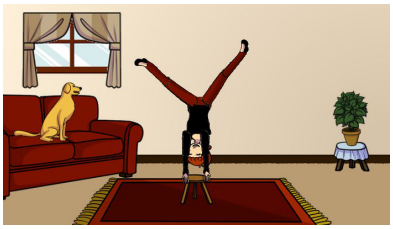
Both SYNTH-VQA and COCO-VQA come in both open-ended and multiple-choice formats. The multiple-choice format contains all the same QA pairs, but it also contains 18 different choices that are comprised of

- **The Correct Answer**, which is the most frequent answer given by the ten annotators.
- **Plausible Answers**, which are three answers collected from annotators without looking at the image.
- **Popular Answers**, which are the top ten most popular answers in the dataset.
- **Random Answers**, which are randomly selected correct answers for other questions.

Due to diversity and size of the dataset, COCO-VQA has been widely used to evaluate algorithms. However, there are multiple problems with the dataset. COCO-VQA has a large variety of questions, but many of them can be accurately answered without using the

(a) Q: Would you like to fly in that? GT: yes (4x), no (6x). The VQA Dataset contains subjective questions that are prone to cause disagreement between annotators and also clearly lack a single objectively correct answer.

(b) Q: What color are the trees? GT: green. There are 73 total questions in the dataset asking this question. For 70 of those questions, the majority answer is green. Such questions can be often answered without information from the image.



(c) Q: Why would you say this woman is strong? GT: yes (5x), can lift up on arms, headstand, handstand, can stand on her head, she is standing upside down on stool. Questions seeking descriptive or explanatory answers can pose significant difficulty in evaluation.

Figure 2.4: Open ended QA pairs from The VQA Dataset for both real and abstract images.

image due to language biases. Relatively simple image-blind algorithms have achieved 49.6% accuracy on COCO-VQA using the question alone [30]. The dataset also contains many subjective, opinion-seeking questions that do not have a single objective answer (see Figure 2.4). Similarly, many questions seek explanations or verbose descriptions. An example of this is given in Figure 2.4c, which also shows unreliability of human annotators as the most popular answer is 'yes' which is completely wrong for the given question. These complications are reflected by inter-human agreement on this dataset, which is about 83%. Several other practical issues also arise out of the dataset's biases. For example, 'yes/no' answers span about 38% of all questions, and almost 59% of them are answered with 'yes.' Combined with the evaluation metric used with COCO-VQA (see Section 2.4), these biases can make it difficult to assess whether an algorithm is truly solving the VQA problem using solely this dataset. We discuss this further in Section 2.4.

### The VQA Dataset - Version 2

As mentioned in preceding section, COCO-VQA (also known as VQAv1) has multiple kinds of language bias, including some questions being heavily correlated with specific answers. In the second version of the dataset, called VQAv2 [80], the authors endeavor to mitigate this kind of language bias by collecting complementary images per question that result in different answers. For example, for a question 'what color is the cat', the authors ensure that there are at least two images with different answer to the same question. This ensures that the correct answer is difficult to guess simply due to language-bias. However other kinds of bias are still present, *e.g.* , complex reasoning questions are rare compared to simple detection questions. Both versions of the VQA dataset have been widely used and VQAv2 has supplanted the VQAv1 as the de facto benchmark for natural image VQA

in recent years.

### 2.3.4 FM-IQA

The Freestyle Multilingual Image Question Answering (FM-IQA) dataset is another dataset based on COCO [72]. It contains human generated answers and questions. The dataset was originally collected in Chinese, but English translations have been made available. Unlike COCO-QA and DAQUAR, this dataset also allowed for answers to be full sentences. This makes automatic evaluation with common metrics intractable. For this reason, the authors suggested using human judges for evaluation, where the judges are tasked with deciding whether or not the answer is provided by a human or not as well as assessing the quality of an answer on a scale of 0–2. This approach is impractical for most research groups and makes developing algorithms difficult. We further discuss the importance of automatic evaluation metrics in Section 2.4.

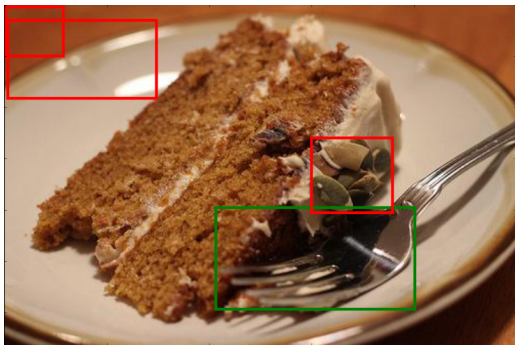
### 2.3.5 Visual Genome

Visual Genome [74] consists of 108,249 images that occur in both YFCC100M [81] and COCO images. It contains 1.7 million QA pairs for images, with an average of 17 QA pairs per image. As of this article, Visual Genome is the largest VQA dataset. Because it was only recently introduced, no methods have been evaluated on it beyond the baselines established by the authors.

Visual Genome consists of six types of 'W' questions: `What, Where, How, When, Who, and Why`. Two distinct modes of data collection were used to make the dataset. In the free-form method, annotators were free to ask any question about an image. However, when asking free-form questions, human annotators tend to ask similar questions about an image's holistic content, e.g., asking 'How many horses are there?' or 'Is it sunny?' This can promote bias in the kinds of questions asked. The creators of Visual Genome combated this by also prompting workers to ask questions about specific image regions. When using this region-specific method, a worker might be prompted to ask a question about a region of an image containing a fire hydrant. Region-specific question prompting was made possible using Visual Genome's descriptive bounding-box annotations. An example of region bounding boxes and QA pairs from Visual Genome are shown in Figure 2.5a.

Visual Genome has much greater answer diversity compared to other datasets, which is shown in Figure 2.6. The 1000 answers that occur most frequently in Visual Genome only cover 65% of all answers in the dataset, whereas they cover 82% for COCO-VQA and 100% for DAQUAR and COCO-QA. Visual Genome's long-tailed distribution is also observed in the length of the answers. Only 57% of answers are single words, compared to 88% of answers in COCO-VQA, 100% of answers in COCO-QA, and 90% of answers

(a) Example image from the Visual Genome dataset along with annotated image regions. This figure is taken from [74].
Free form QA: What does the sky look like?
Region based QA: What color is the horse?



(b) Example of the pointing QA task in Visual7W [73]. The bounding boxes are the given choices. Correct answer is shown in green
Q: Which object can you stab food with?

Figure 2.5: Visual7W is a subset of Visual Genome. Apart from the pointing task, all of the questions in Visual7W are sourced from Visual Genome data. Visual Genome, however, includes more than just QA pairs, such as region annotations.

in DAQUAR. This diversity in answers makes open-ended evaluation significantly more challenging. Moreover, since the categories themselves are required to strictly belong to one of the six 'W' types, the diversity in answer may at times artificially stem simply from variations in phrasing which could be eliminated by prompting the annotators to choose more concise answers. For example, *Where are the cars parked?* can be answered with 'on the street' or more concisely with 'street.'

Visual Genome has no binary (yes/no) questions. The dataset creators argue that this will encourage using more complex questions. This is in contrast to The VQA Dataset, where 'yes' and 'no' are the more frequent answers in the dataset. We discuss this issue further in Section 2.6.4.

### 2.3.6 Visual7W

The Visual7W dataset is a subset of Visual Genome. Visual7W contains 47,300 images from Visual Genome that are also present in COCO. Visual7W is named after the seven categories of questions it contains: `What`, `Where`, `How`, `When`, `Who`, `Why`, and `Which`. The dataset consists of two distinct types of questions. The 'telling' questions are identical to Visual Genome questions, and the answer is text-based. The 'pointing' questions are the ones that begin with 'Which,' and for these questions the algorithm has to select the correct bounding box among alternatives. An example pointing question is shown in Figure 2.5b.

Visual7W uses a multiple-choice answer framework as the standard evaluation, with

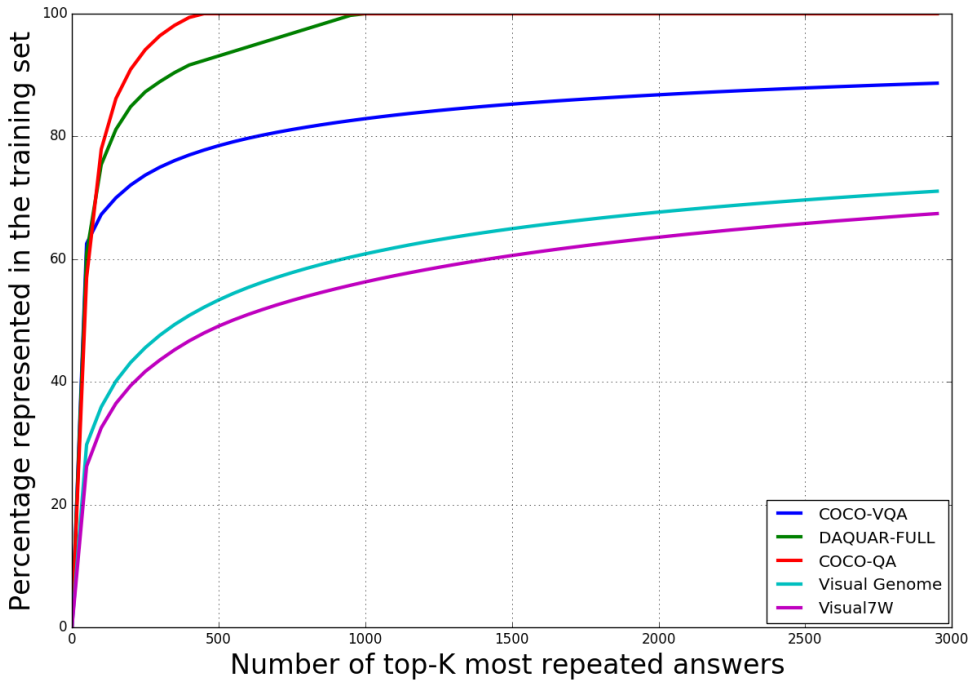



Figure 2.6: This graph shows the long-tailed nature of answer distributions in newer VQA datasets. For example, choosing the 500 most repeated answers in the training set would cover a 100% of all possible answers in COCO-QA but less than 50% in the Visual Genome dataset. For classification based frameworks, this translates to training a model with more output classes.

four possible answers being made available to an algorithm during evaluation. To make the task challenging, the multiple-choices consist of answers that are plausible for the given question. Plausible answers are collected by prompting annotators to answer the question without seeing the image. For pointing questions, the multiple-choice options are four plausible bounding boxes surrounding the likely answer. Like Visual Genome, the dataset does not contain any binary questions.

### 2.3.7 SHAPES

While the other VQA datasets contain either real or synthetic scenes, the SHAPES dataset [82] consists of shapes of varying arrangements, types, and colors. Questions are about the attributes, relationships, and positions of the shapes. This approach enables the creation

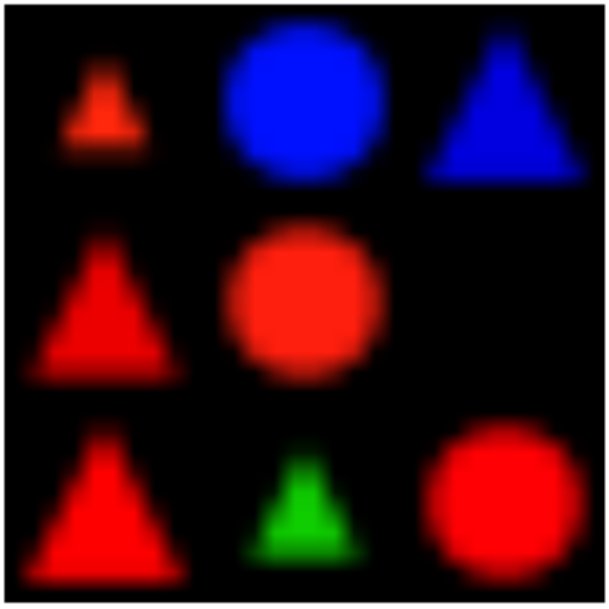


Figure 2.7: Example image from the SHAPES dataset. Questions in the SHAPES dataset [82] include counting (How many triangles are there?), spatial reasoning (Is there a red shape above a circle?), and inference (Is there a blue shape red?)

of a vast amount of data, free of many of the biases that plague other datasets to varying degrees.

SHAPES consists of 244 unique questions, with every question asked about each of the 64 images in the dataset. Unlike other datasets, this means it is completely balanced and free of bias. All questions are binary, with yes/no answers. Many of the questions require positional reasoning about the layout and properties of the shapes. While, SHAPES cannot be a substitute for using real-world imagery, the idea behind it is extremely valuable. An algorithm that cannot perform well on SHAPES, but performs well on other VQA datasets may indicate that it is only capable of analyzing images in a limited manner.

### 2.3.8 CLEVR

**CLEVR** [83] is a synthetically generated dataset, consisting of visual scenes with simple geometric shapes, designed to test 'compositional language and elementary visual reasoning.' CLEVR is made of over 700,000 QA pairs for 70,000 synthetically generated images. CLEVR specifically tests for multi-step compositional reasoning that is very

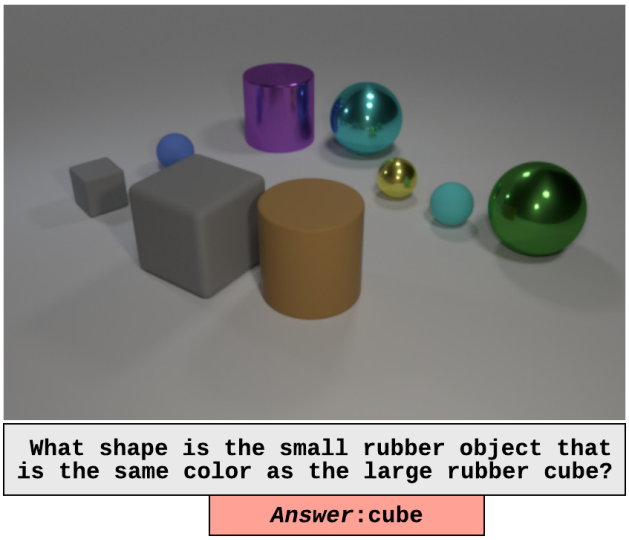



Figure 2.8: Example image from the CLEVR dataset. Questions in the CLEVR include 'querying attribute,' 'comparing attributes,' 'existence,' 'counting,' and 'integer comparison.'

rarely encountered in natural image VQA datasets. Hence, CLEVR's questions often require long chains of complex reasoning. To enable fine-grained evaluation of reasoning abilities, CLEVR's questions are categorized into five tasks: 'querying attribute,' 'comparing attributes,' 'existence,' 'counting,' and 'integer comparison.' Figure 2.8 shows a sample CLEVR image with corresponding question-answer pair.

Since all of the questions are programmatically generated, the **CLEVR-Humans** [84] dataset was later added to supplement the dataset with human-generated questions for CLEVR scenes to test generalization to free-form questions. In another extension, **CLEVR-CoGenT** tests the ability to handle unseen concept composition and remember old concept combinations. It has two splits: CoGenT-A and CoGenT-B, with mutually exclusive shape+color combinations. If models trained on CoGenT-A perform well on CoGenT-B without fine-tuning, it indicates generalization to novel compositions. If models fine-tuned on CoGenT-B still perform well on CoGenT-A, it indicates the ability to remember old concept combinations.

## 2.4 Evaluation Metrics for VQA

VQA has been posed as either an open-ended task, in which an algorithm generates a string to answer a question, or as a multiple-choice question where it picks among choices.

Figure 2.9: Simple questions can also evoke diverse answers from annotators in COCO-VQA. Q: Where is the dog? A: 1) eating out of his bowl; 2) on floor; 3) feeding station; 4) by his food; 5) inside; 6) on floor eating out of his dish; 7) floor; 8) in front of gray bowl, to right of trash can; 9) near food bowl; 10) on floor

For multiple-choice, simple accuracy is often used to evaluate, with an algorithm getting an answer right if it makes the correct choice. For open-ended VQA, simple accuracy can also be used. In this case, an algorithm's predicted answer string must exactly match the ground truth answer. However, accuracy can be too stringent because some errors are much worse than others. For example, if the question was 'What animals are in the photo?' and a system outputs 'dog' instead of the correct label 'dogs,' it is penalized just as strongly as it would be if it output 'zebra.' Questions may also have multiple correct answers, e.g., 'What is in the tree?' might have 'bald eagle' listed as the correct ground truth answer, so a system that outputs 'eagle' or 'bird' would be penalized just as much as if it had output 'yes' as the answer. Due to these issues, several alternatives to exact accuracy have been proposed for evaluating open-ended VQA algorithms.

One way to handle this problem is to use the Wu-Palmer Similarity (WUPS) index [85], which is used to evaluate VQA systems in [71], [25] and [78]. WUPS ranges between 0 through 1, where 1.0 is perfect match between semantic meaning of two words being compared. It does this by finding the least common subsumer between two semantic senses and assigning scores based on how far back the semantic tree needs to be traversed to find the common subsumer. Using WUPS, semantically similar, but non-identical, words are penalized relatively less. This measure can be used to relax the

[11] We were unable to retrieve the English version of the dataset from provided download link.

stringent requirement of accuracy measure which unnecessarily penalizes semantically similar answers.Following our earlier example, 'bald eagle' and 'eagle' have similarity of 0.96, whereas 'bald eagle' and 'bird' have similarity of 0.88. WUPS calculates similarity between two specific word senses but each word can have multiple senses. In this regard, [25] suggest using a metric that considers similarity between all possible combinations between a set of word senses produced from two words being compared and returns maximum similarity between them.

However, WUPS tends to assign relatively high scores to even distant concepts, e.g., 'raven' and 'writing desk' have a WUPS score of 0.4. To remedy this, [25] proposed to threshold WUPS scores, where a score that is below a threshold will be scaled down by a factor. A threshold of 0.9 and scaling factor of 0.1 was suggested by [25]. This modified WUPS metric is the standard measure used for evaluating performance on DAQUAR and COCO-QA, in addition to simple accuracy. Besides WUPS, there are other ways to measure the semantic distance between words, e.g., measuring semantic distance in a distributed representation of words [86]. This has been studied in automatic evaluation of captioning [87] but is not widely studied for evaluation of VQA algorithms.

There are two major shortcomings to WUPS that make it difficult to use. First, despite using a thresholded version of WUPS, certain pairs of words are lexically very similar but carry vastly different meaning. This is particularly problematic for questions about object attributes, such as color questions. For example, if the correct answer was 'white' and the predicted answer was 'black,' the answer would still receive a WUPS score of 0.91, which seems excessively high. Another major problem with WUPS is that it only works with rigid semantic concepts, which are almost always single words. WUPS cannot be used for phrasal or sentence answers that are occasionally found in The VQA Dataset and in much of Visual7W.

An alternative to relying on semantic similarity measures is to have multiple independently collected ground truth answers for each question, which was done for The VQA Dataset [12] and DAQUAR-consensus [78]. For DAQUAR-consensus, an average of five human annotated ground truth answers per question were collected. The dataset's creators proposed two ways to use these answers, which they called average consensus and min consensus. For average consensus, the final score is weighted toward preferring the more popular answer provided by the annotators. For min consensus, the answer needs to agree with at least one annotator.

For The VQA Dataset, annotators generated ten answers per question. These are used with a variation of the accuracy metric, which is given by

$$Accuracy_{VQA} = \min(\frac{n}{3}, 1), \tag{2.1}$$

where $n$ is the total number of annotators that had the same answer as the algorithm. Using this metric, if the algorithm agrees with three or more annotators then it is awarded

a full score for a question. Although this metric helps greatly with the ambiguity problem, substantial problems remain, especially with the COCO-VQA portion of the dataset, which we study further in the next few paragraphs[2].

Using $Accuracy_{VQA}$, the inter-human agreement on COCO-VQA is only 83.3%. It is impossible for an algorithm to achieve 100% accuracy. Inter-human agreement is especially poor for 'Why' questions, with over 59% of these questions having less than three annotators giving exactly the same answer. This makes it impossible to get a full score on these questions. Lack of inter-human agreement can also be seen in simpler, more straightforward questions (see Figure 2.9). In this example, if a system predicts any of the 10 answers, it will be awarded a score of at least 1/3. In several cases, the answers provided by annotators consist complete antonyms (e.g., left and right).

In many other cases, $Accuracy_{VQA}$ leads to multiple correct answers for a question that are in direct opposition to each other. For example, in COCO-VQA more than 13% of the 'yes/no' answers have both 'yes' and 'no' repeated by more than three annotators. Either answering 'yes' or 'no' would receive the highest possible score. Even if eight annotators answered 'yes,' if two answered 'no' then an algorithm would still receive a score of 0.67 for the question. The weight of the majority does not play a role in evaluation.

These problems can result in the scores being inflated. For example, answering 'yes' to all yes/no questions should ideally have a score of around 50% for those questions. However, using $Accuracy_{VQA}$, the score is 71%. This is partially due to the dataset being biased, with the majority answer for these questions being 'yes' 58% of the time, but a score of 71% is excessively inflated.

Evaluating the open-ended responses of VQA systems is made simpler when the answers consist of one word answers. This occurs in 87% of COCO-VQA questions, 100% of COCO-QA questions, and 90% of DAQAUR questions. The possibility of multiple correct answers increases greatly when answers need to be multiple words. This occurs frequently in FM-IQA, Visual7W, and Visual Genome, e.g., 27% of Visual7W answers have three or more words. In this scenario, metrics such as $Accuracy_{VQA}$ are unlikely to help score predicted answers to ground truth answers in open-ended VQA.

The creators of FM-IQA [72] suggested using human judges to assess multi-word answers, but this presents a number of problems. First, using human judges is an extremely demanding process in terms of time, resources, and expenses. It would make it difficult to iteratively improve a system by measuring how changing the algorithm altered performance. Second, human judges need to be given criteria for judging the quality of an answer. The creators of FM-IQA proposed two metrics for human judges. The first is to determine whether the answer was produced by a human or not, regardless of the answer's

[2]Note that our analysis for COCO-VQA was only done on the train and validation portions of the dataset, because the test answers are not publicly available.

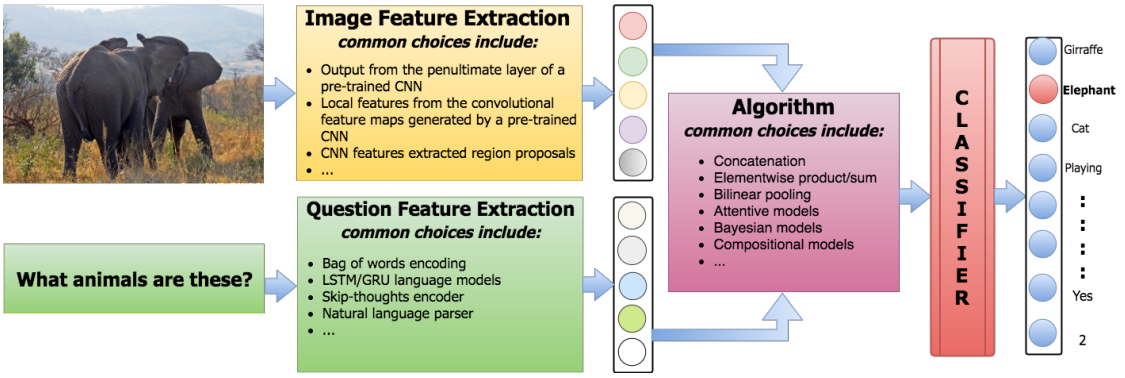



Figure 2.10: Simplified illustration of the classification based framework for VQA. In this framework, image and question features are extracted, and then they are combined so that a classifier can predict the answer. A variety of feature extraction methods and algorithms for combining these features have been proposed, and some of the more common approaches are listed in their respective blocks in the figure. Full details are presented in Section 2.5.

correctness. This metric alone may be a poor indicator of a VQA system's abilities and could potentially be manipulated. The second metric is to rate an answer on a 3-point scale of totally wrong (0), partially correct (1), and perfectly correct (2).

An alternative to using judges for handling multi-word answers is to use a multiple-choice paradigm, which is used by part of The VQA Dataset, Visual7W, and Visual Genome. Instead of generating an answer, a system only needs to predict which of the given choices is correct. This greatly simplifies evaluation, but we believe that unless it is used carefully, multiple-choice is ill-suited for VQA because it undermines the effort by allowing a system to peek at the correct answer. We discuss this issue in Section 2.6.5.

The best way to evaluate a VQA system is still an open question. Each evaluation method has its own strengths and weaknesses (see Table 2.2 for a summary). The method to use depends on how the dataset was constructed, the level of bias within it, and available resources. Considerable work needs to be done to develop better tools for measuring the semantic similarity of answers and for handling multi-word answers.

## 2.5 Algorithms for VQA

A large number of VQA algorithms have been proposed in the past three years. In general, VQA algorithms have three sub-systems: 1) visual processing, 2) language processing, and 3) multi-modal integration to produce an answer.

For visual processing, almost all algorithms use CNN features. Typically, CNNs that are pre-trained on ImageNet are used, with common examples being VGGNet [3],

ResNet [4], and GoogLeNet [100]. Shallow trained-from-scratch CNNs are also used for synthetic scene datasets [1]. Until 2017, most algorithms for natural scenes used CNN features directly; however, more recent algorithms have switched to using CNN region proposal features [101]. Another recent trend is the use of graph-based representations for image retrieval [20], image generation [102], VQA [103], and semantic knowledge incorporation [103], due to their intuitiveness and suitability for symbolic reasoning.

For language representation, most VQA systems process words using recurrent neural networks (RNNs). For tasks that take queries as input, word tokens fed to the RNN are commonly learned as vector embeddings in an end-to-end manner with the network being trained on a downstream-task [16, 104, 105]. Recent VQA systems leverage distributed representations of words trained on large corpora of natural language text. Common choices include word2vec [86], GloVe [106] and fasttext [107]. A few approaches have incorporated explicit syntax and semantic information from language, such as part-of-speech based semantic parsing [16] and dependency trees [108]; however, distributed vector representations remain the dominant language representation for most recent systems.

To generate an answer via multimodal reasoning, the most common approach is to treat VQA as a classification problem. In this framework, the image and question features are the input to the classification system and each unique answer is treated as a distinct category. As illustrated in Figure 2.10, the featurization scheme and the classification system can take widely varied forms. These systems differ significantly in how they integrate the question and image features. Some examples include:

- Combining the image and question features using simple mechanisms, e.g., concatenation, elementwise multiplication, or elementwise addition, and then giving them to a linear classifier or a neural network [12, 30, 72, 76],
- Combining the image and question features using bilinear pooling or related schemes in a neural network framework [88, 95, 109, 110],
- Having a classifier that uses the question features to compute spatial attention maps for the visual features or that adaptively scales local features based on their relative importance [90, 91, 93, 101, 104, 111],
- Using Bayesian models that exploit the underlying relationships between question-image-answer feature distributions [25, 30], and
- Using the question to break the VQA task into a series of sub-problems [82, 92].

In later subsections, we describe each of these classification-based approaches in detail.

While the classification framework is used by most open-ended VQA algorithms, this approach can only generate answers that are seen during training, prompting some to explore alternatives. In [72] and [78] an LSTM is used to produce multi-word answer one word at a time. However, the answer produced is still limited to words seen during training. For multiple-choice VQA, [112] and [111] proposed treating VQA as a ranking

problem, where a system is trained to produced a score for each possible multiple-choice answer, question, and image trio, and then it selects the highest scoring answer choice.

In the following subsections, we group VQA algorithms based on their common themes. Results on DAQUAR, COCO-QA, and COCO-VQA for these methods are given in Table 2.3, in increasing order of performance. In Table 2.3, we report plain accuracy for DAQUAR and COCO-QA, and we report $Accuracy_{VQA}$ for COCO-VQA. Table 2.4 breaks down the results for COCO-VQA based on the techniques used in each paper. A similar detailed breakdown of impact of different features and other network choices for the VQAv2 dataset can be found in [113].

### 2.5.1 Baseline Models

Baseline methods help determine the difficulty of a dataset, and establish the minimal level of performance that a more sophisticated algorithms should exceed. For VQA, the simplest baselines are random guessing and guessing the most repeated answers. A widely used baseline classification system is to apply a linear or non-linear, e.g., multi-layer perceptron (MLP), classifier to the image and question features after they have been combined into a single vector [12, 30, 76]. Common methods to combine the features include concatenation, the elementwise product, or the elementwise sum. Combining these schemes has also been explored and can lead to improved results [109].

A variety of featurization approaches have been used with baseline classification frameworks. In [76], the authors used a bag-of-words to represent the question and CNN features from GoogLeNet for the visual features. They then fed concatenation of these features into a multi-class logistic regression classifier. Their approach worked well, surpassing the previous baseline on COCO-VQA, which used a theoretically more powerful model, an LSTM, to represent the question [12]. Similarly, [30] used skip-thought vectors [114] for question features and ResNet-152 to extract image features. They found that an MLP model with two hidden layers trained on these off-the-shelf features worked well for all datasets. However, in their work a linear classifier outperformed the MLP model on smaller datasets, likely due to the MLP model overfitting.

Several VQA algorithms have used LSTMs to encode questions. In [12], an LSTM encoder acting on a one-hot encoding of the sentence was used to represent question features, and GoogLeNet was used for image features. The dimensionality of the CNN features was reduced to match the dimensionality of the LSTM encoding, and then the Hadamard product of these two vectors was used to fuse them together. The fused vector was used as input to an MLP with two hidden layers. In [78], an LSTM model was fed an embedding of each word sequentially with CNN features concatenated to it. This continued until the end of the question was reached. The subsequent time-steps were used to generate a list of answers. A related approach was used in [71], where an LSTM was fed CNN features during the first and last time-steps, with word features in between. The

image features acted as the first and last words in the sentence. The LSTM network was followed by a softmax classifier to predict the answer. A similar approach was used in [72], but the CNN image features were only fed into the LSTM at the end of the question and instead of a classifier, another LSTM was used to generate the answer one word at a time.

### 2.5.2 Bayesian and Question-Aware Models

VQA requires drawing inferences and modeling relationships between the question and the image. Once the questions and images are featurized, modeling co-occurrence statistics of the question and image features can be helpful for drawing inferences about the correct answers. Two major Bayesian VQA frameworks have explored modeling these relationships. In [25], the first Bayesian framework for VQA was proposed. The authors used semantic segmentation to identify the objects in an image and their positions. Then, a Bayesian algorithm was trained to model the spatial relationships of the objects, which was used to compute each answer's probability. This was the earliest known algorithm for VQA, but its efficacy is surpassed by simple baseline models. This is partially due to it being dependent on the results of the semantic segmentation, which was imperfect.

We proposed a very different Bayesian model in [30], which we describe in Chapter 3. This model exploited the fact that the type of answer can be predicted using solely the question. For example, 'What color is the flower?' would be assigned as a color question by the model, essentially turning the open-ended problem into a multiple-choice one. To do this, the model used a variant of quadratic discriminant analysis, which modeled the probability of image features given the question features and the answer type. ResNet-152 was used for the image features, and skip-thought vectors were used to represent the question.

### 2.5.3 Attention Based Models

Using global features alone may obscure task-relevant regions of the input space. Attentive models attempt to overcome this limitation. These models learn to 'attend' to the most relevant regions of the input space. Attention models have shown great successes in other vision and NLP tasks, such as object recognition [115], captioning [50] and machine translation [116, 117].

In VQA, numerous models have used spatial attention to create region-specific CNN features, rather than using global features from the entire image. Fewer models have also explored incorporating attention into the text representation. The basic idea behind all these models is that certain visual regions in an image and certain words in a question are more informative than others for answering a given question. For example, for a system answering 'What color is the umbrella?' the image region containing the umbrella is more

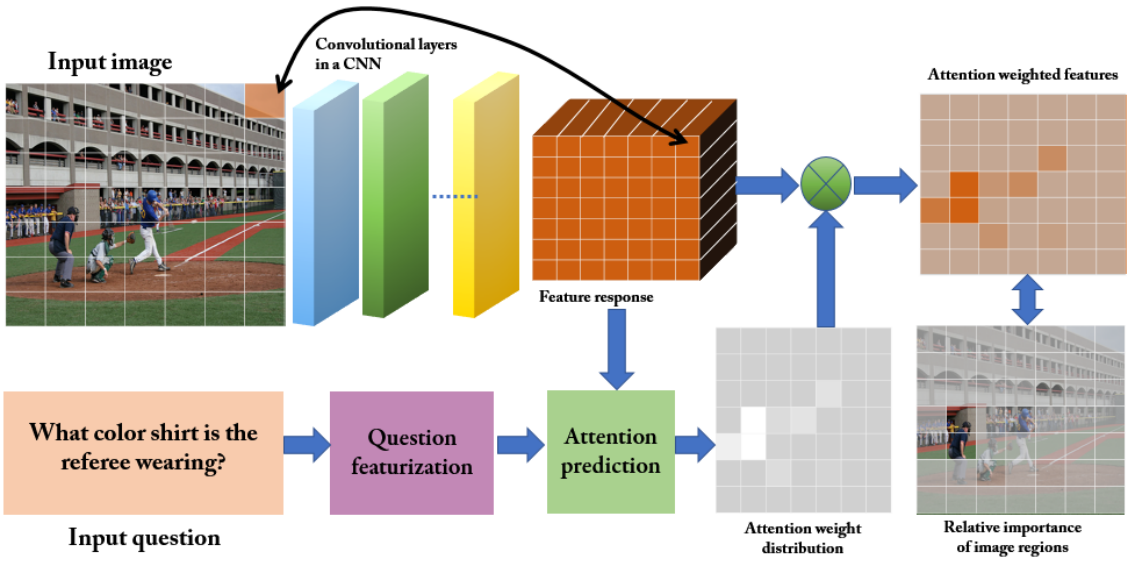



Figure 2.11: This figure illustrates a common way to incorporate attention into a VQA system. A convolutional layer in a CNN outputs a $K \times K \times N$ tensor of feature responses, corresponding to $N$ feature maps. One way to apply attention to this representation is by suppressing or enhancing the features at different spatial locations. Using the question features with these local image features, a weighting factor for each grid location can be computed that determines the spatial location's relevance to the question, which can then be used to compute attention-weighted image features.

informative than other image regions. Similarly, 'color' and 'umbrella' are the textual inputs that need to be addressed more directly than the others. Global image features, e.g., the last hidden layer of a CNN, and global text features, e.g., bag-of-words, skip-thoughts etc. may not be granular enough to address region specific questions.

Before using spatially attentive mechanisms, an algorithm must represent the visual features at all spatial regions, instead of solely at the global level. Then, local features from relevant regions can be given higher prominence based on the question asked. There are two common ways to achieve local feature encoding. As shown in Figure 2.11, one way to do this is to impose a uniform grid over all image locations, with the local image features present at each grid location. This is often done by operating on the last CNN layer prior to the final spatial pooling that flattens the features. More recent algorithms have switched to using CNN region proposal features [101] from an network trained for object detection. The relevance of each image region (grid features or region proposals) is then determined by the question. While multiple papers have focused on using spatial visual attention for VQA [88, 90, 91, 93, 94, 96, 97, 111], there are significant differences

among these methods.

The Focus Regions for VQA [111] and Focused Dynamic Attention (FDA) models [93] both used Edge Boxes [118] to generate bounding box region proposals for images. In [111], a CNN was used to extract features from each of these boxes. The input to their VQA system consisted of these CNN features, question features, and one of the multiple choice answers. Their system was trained to produced a score for each multiple-choice answer, and the highest scoring answer was selected. The score is calculated using a weighted average of scores from each of the regions where the weights are simply learned by passing the dot product of regional CNN feature and question embedding to a fully connected layer.

In FDA [93], the authors proposed to only use the region proposals that have the objects mentioned in the question. Their VQA algorithm requires as input a list of bounding boxes with their corresponding object label. During training, the object labels and bounding boxes are obtained from COCO annotations. During test, the labels are obtained by classifying each bounding box using ResNet [4]. Subsequently, word2vec [86] was used to compute the similarity between words in the question and the object labels assigned to each of the bounding boxes. Any box with a score greater than 0.5 is successively fed into an LSTM network. At the last time-step, global CNN features from the entire image are also fed into the network, giving it access to both global and local features. A separate LSTM was also used as the question representation. The output from these two LSTMs are then fed into a fully connected layer that is fed to a softmax classifier to produce the answer predictions.

In contrast to using region proposals, the Stacked Attention Network (SAN) [91] and the Dynamic Memory Network (DMN) [94] models both used visual features from the spatial grid of a CNN's feature maps (see Figure 2.11). Both [91] and [94] used the last convolutional layer from VGG-19 with $448 \times 448$ images to produce a $14 \times 14$ filter response map with 512 dimensional features at each grid location.

In SAN [91], an attention layer is specified by a single layer of weights that uses the question and the CNN feature map with a softmax activation function to compute the attention distribution across image locations. This distribution is then applied to the CNN feature map to pool across spatial feature locations using a weighted sum, which generates a global image representation that emphasizes certain spatial regions more than others. This feature vector is then combined with a vector of question features to create a representation that can be used with a softmax layer to predict the answer. They generalized this approach to handle multiple (stacked) attention layers, enabling the system to model complex relationships among multiple objects in an image. In a similar approach [101], combines bottom-up and top-down attention mechanisms. This work uses region proposals from from Faster R-CNN [6] network for object detection. Since these proposal are generated based on their "object-ness", the authors call this the "bottom-up" pathway. The top-down mechanism predicts an attention distribution over those proposals based on

the question, which is computed in similar manner to [91].

A similar attentive mechanism was used in the Spatial Memory Network [90] model, where spatial attention is produced by estimating the correlation of image patches with individual words in the question. This word-guided attention is used to predict an attention distribution, which is then used to compute the weighted sum of the visual features embedding across image regions. Two different models were then explored. In the one-hop model, the features encoding the entire question are combined with the weighted visual features to predict the answer. In the two-hop model, the combination of the visual and question features is looped back into the attentive mechanism for refining the attention distribution.

Another approach that incorporated spatial attention using CNN feature maps is presented in [94]. To do this, they used a modified Dynamic Memory Network (DMN) [119]. A DMN consists of an input module, an episodic memory module, and an answering module. DMNs have been used for text based QA, where each word in a sentence is fed into a recurrent neural network and the output of the network is used to extract 'facts.' Then, the episodic memory module makes multiple passes over a subset of these facts. With each pass, the internal memory representation of the network is updated. An answering module uses the final state of the memory representation and the input question to predict an answer. To use a DMN for VQA, they used visual facts in addition to text. To generate visual facts, the CNN features at each spatial grid location are treated as words in a sentence that are sequentially fed into a recurrent neural network. The episodic memory module then makes passes through both text and visual facts to update its memory. The answering module remains unchanged.

The Hierarchical Co-Attention model [96] applies attention to both the image and question to jointly reason about the two different streams of information. The model's approach to visual attention is similar to the method used in Spatial Memory Network [90]. In addition to visual attention, this method uses a hierarchical encoding of the question, in which the encoding occurs at the word level (using a one-hot encoding), at the phrase level (using bi- or tri-gram window size), and at the question level (using the final timestep of an LSTM network). Using this hierarchical question representation, the authors proposed to use two different attentive mechanisms. The parallel co-attention approach simultaneously attended to both the question and image. The alternative co-attention approach alternated between attending to the question or the image. This approach allowed the relevance of words in the question and the relevance of specific image regions to be determined by each other. The answer prediction is made by recursively combining the co-attended features from all three levels of the question hierarchy.

Using joint attention for image and question features was also explored in [98]. The main idea is to allow image and question attention to guide each other, directing attention to relevant words and visual regions simultaneously. To achieve this, visual and question input are jointly represented by a memory vector that is used to simultaneously predict

attention for both question and image features. The attentive mechanism computes updated image and question representations, which are then used to recursively update the memory vector. This recursive memory update mechanism can be repeated $K$ times to refine the attention in multiple steps. The authors' found that a value of $K = 2$ worked best for COCO-VQA.

In a recent study, Bilinear Attention Network (BAN) [104] proposes an even richer interaction and attention over visual and textual modalities by considering interactions between all image regions with all question words. Unlike dual-attention mechanisms [98], BAN handles interactions between all channels.

### 2.5.4 Bilinear Pooling Methods

VQA relies on jointly analyzing the image and the question. Early models did this by combining their respective features using simple methods, e.g., concatenation or using an element-wise product between the question and image features, but more complex interactions would be possible with an outer-product between these two streams of information. Similar ideas were shown to work well for improving fine-grained image recognition [120]. Below, we describe the two most prominent VQA methods that have used bilinear pooling [88, 99].

In [88], Multimodal Compact Bilinear (MCB) pooling was proposed as a novel method for combining image and text features in VQA. This idea is to approximate the outer-product between the image and text features, allowing a deeper interaction between the two modalities, compared to other mechanisms, e.g., concatenation or element-wise multiplication. Rather than doing the outer-product explicitly, which would be very high dimensional, MCB does the outer-product in a lower dimensional space. This is then used to predict which spatial features are relevant to the question. In a variation of this model, a soft-attention mechanism, similar to the method in [91], was also used, with the only major change being the use of MCB for combining text and question features instead of element-wise multiplication in [91]. This combination yielded very good results on COCO-VQA, and it was the winner of the 2016 VQA Challenge workshop.

In [99], the authors' argued that MCB is too computationally expensive, despite using an approximate outer-product. Instead, they proposed to use a multi-modal low-rank bilinear pooling (MLB) scheme that uses the Hadamard product and a linear mapping to achieve approximate bilinear pooling. When used with a spatial visual attention mechanism, MLB rivaled MCB at VQA, but with reduced computational complexity and using a neural network with fewer parameters.

### 2.5.5 Compositional VQA Models

In VQA, questions often require multiple steps of reasoning to answer properly. For example, questions like 'What is to the left of the horse?' can involve first finding the horse, and then naming the object to the left of it. Two compositional frameworks have been proposed for VQA that attempt to tackle solving VQA in a series of sub-steps [82, 92, 97]. The Neural Module Network (NMN) [82, 92] framework uses external question parsers to find the sub-task in the question whereas Recurrent Answering Units (RAU) [97] is trained end-to-end and sub-tasks can be implicitly learned.

NMN is an especially interesting approach to VQA [82, 92]. The NMN framework treats VQA as a sequence of sub-tasks that are carried out by separate neural sub-networks. Each of the sub-network performs a single well-defined task, e.g., the `find[X]` module produces a heat map for the presence of certain object. Other modules include `describe`, `measure`, and `transform`. These modules then must be assembled into a meaningful layout. Two methods have been explored for inferring the required layout. In [82], a natural language parser is used on the input question to both find the sub-tasks in the question and to infer the required layout of the sub-tasks that when executed in sequence would produce an answer to the given question [82]. For example, answering 'What color is the tie?' would involve executing the `find[tie]` module followed by the `describe[color]` module, which generates the answer. In [92], the same group explored using algorithms to dynamically select the best layout for the given question from a set of automatically generated layout candidates.

The Memory, Attention and Composition (MAC) network [121] uses computational cells that automatically learn to perform attention-based reasoning. Unlike, modular networks [82, 92, 122] that require pre-defined modules to perform pre-specified reasoning functions, MAC learns reasoning mechanisms directly from the data. Each MAC cell maintains a control state representing the reasoning operation and a memory state that is the result of the reasoning operation. It has a computer-like architecture with read, write and control units. MAC was evaluated on the CLEVR dataset and reports significant improvements on the challenging counting and numerical comparison tasks. Similarly, compositional reasoning can also be achieved by capturing pairwise interactions between V&L representations as explored by relational networks (RN) [1], which is also evaluated on the CLEVR dataset.

The RAU model [97] can implicitly perform compositional reasoning without depending on an external language parser. In their model, they used multiple self-contained answering units that can solve VQA sub-tasks. These answering units are arranged in recurrent manner. Each answering unit on the chain is equipped with an attentive mechanism derived from [91] and a classifier. The authors' claimed that the inclusion of multiple recurrent answering units allows inferring the answer from a series of sub-tasks solved by each answering unit. However, they did not perform visualization or ablation studies to

show how the answer might get refined in each time-step. This makes it difficult to assess whether progressive refinement and reasoning is occurring or not, especially considering that the complete image and question information is available to all answering units at every time step and that only the output from the first answering unit is used during the test stage.

### 2.5.6 Other Noteworthy Models

Answering questions about images can often require information beyond what can be directly inferred by analyzing the image. Having knowledge about the uses and typical context for the objects present in an image can be helpful for VQA. For example, a VQA system that has access to a knowledge bank could use it to answer questions about particular animals, such as their habitats, colors, sizes, and feeding habits. This idea was explored in [75], and they demonstrated that the knowledge bank improved performance. The external knowledge bases were tailored to general information obtained from DBpedia [123], and it is possible that using a source tailored to VQA could yield greater improvement.

In [89], the authors' incorporated a Dynamic Parameter Prediction layer into the fully connected layers of a CNN. The parameters of this layer are predicted from the question by using a recurrent neural network. This allows the visual features that the model uses to be specific to the question before the final classification step. This approach can be seen as a kind of implicit attentive mechanism in that it modifies the visual input based on the question.

In [95], Multimodal Residual Networks (MRN) were proposed for VQA, which were motivated by the success of the ResNet architecture in image classification. Their system is a modification of ResNet [4] to use both visual and question features in the residual mapping. The visual and question embedding are allowed to have their own residual blocks with skip connections. However, after each residual block the visual data is interweaved with the question embedding. The authors explored several alternate arrangement for constructing the residual architecture with multi-modal input and chose the above network based on performance.

### 2.5.7 What methods and techniques work better?

Although many methods have been proposed for VQA, it is difficult to determine what general techniques are superior. Table 2.4 provides a breakdown of the different algorithms evaluated on COCO-VQA based on the techniques and design choices that they utilize. Table 2.4 also includes ablation models from respective algorithms, whenever possible. The ablation models help us to identify the individual contributions of the design choices made by the authors. The first observation we can make is that ResNet

produces superior performance over VGGNet or GoogLeNet across multiple algorithms. This is evident from the models that use identical setup and only change the image representation. In [97], an increase of 2% was observed by using ResNet-101 instead of the VGG-16 CNN for image features. In [96], they found an increase of 1.3% when making the same change in their model. Similarly, changing VGG-19 to ResNet-152 increased performance by 2.3% in [98]. This clearly shows the importance of better image features. In a recent study, [124] showed that the model described in [91] performed over 8% better when an updated visual feature representation was used. An extensive study of impact of different visual features and several other hyper-parameters for different models can be found in [113].

In general, spatial attention can be used to increase performance for a model. This is shown by experiments in [88] and [96], where the models were evaluated with and without attention, and the attentive version performed better in both cases. However, attention alone does not appear to be sufficient. We further discuss this in Section 2.6.2.

Bayesian and compositional architectures do not significantly improve over comparable models, despite being interesting approaches. In one of our work [30] (described in Chapter 3), the Bayesian model performed competitively only after it was combined with an MLP model. It is unclear whether the increase was due to model averaging or the proposed Bayesian method. Similarly, the NMN models in [82] and [92] do not outperform comparable non-compositional models, e.g., [91]. It is possible that both of these methods perform well on specific VQA sub-tasks, e.g., NMN was shown to be specially helpful for positional reasoning questions in the SHAPES dataset. However, since major datasets do not provide a detailed breakdown of question types, it is not possible to quantify how systems perform on specific question types. Moreover, any improvements on rare question types will have negligible impact on the overall performance score, making it difficult to properly evaluate the benefits of these methods. We further discuss these issues in Section 2.6.3.

## 2.6 Discussion

As shown in Figure 2.12, there has been rapid improvement in the performance of VQA algorithms, but there is still a significant gap between the best methods and humans. It remains unclear whether the improvements in performance come from the mechanisms incorporated into later systems, e.g., attention, or if it is due to other factors. Moreover, it can be difficult to decouple the contributions of text and image data in isolation. There are also numerous challenges to comparing algorithms due to the variations in how they are evaluated. In this section, we discuss each of these issues.

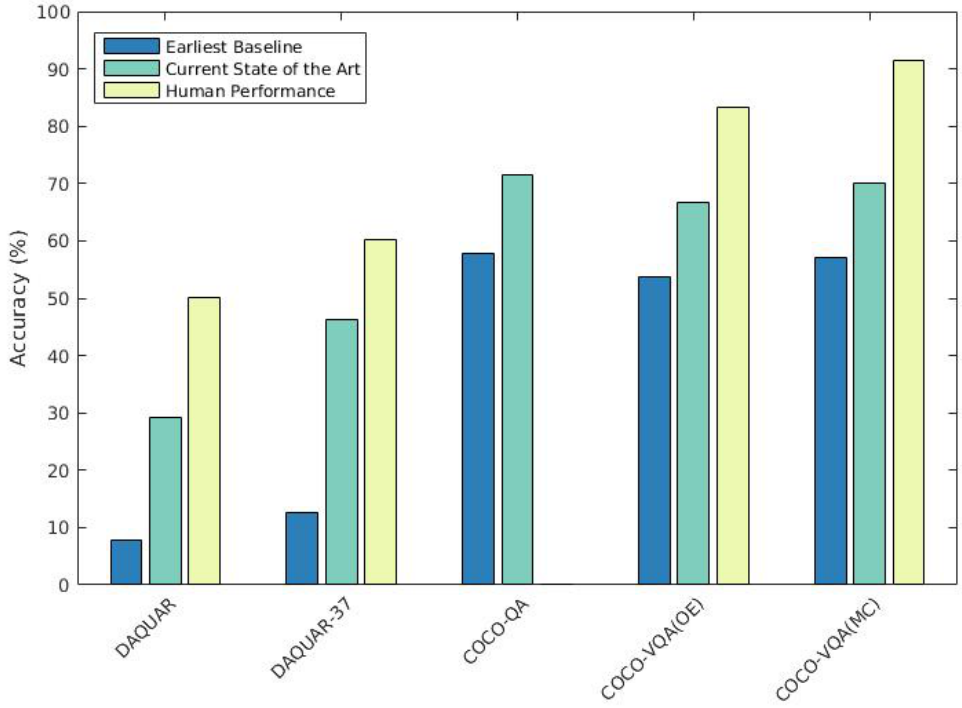



Figure 2.12: Current state-of-the-art results across datasets compared to the earliest baseline and human performance. The earliest baseline refers to the numbers reported by the creators of the datasets and the current state-of-the-art models are chosen from the highest performing methods in Table 2.3. DAQUAR, DAQUAR-37 and COCO-QA report plain accuracy and COCO-VQA reports $Accuracy_{VQA}$.

### 2.6.1 Vision vs. Language in VQA

VQA consists of two distinct data streams that need to be correctly used to ensure robust performance: images and questions. But, do current systems adequately use both vision and language? Ablation studies [12, 30] have routinely shown that question only models perform drastically better than image only models, especially on open-ended COCO-VQA. On COCO-QA, simple image-blind models that use only the question can achieve 50% accuracy with the gain from using the image being comparatively modest [30]. In [30], it was also shown that for DAQUAR-37, using a better language embedding with an image-blind model produced results superior to earlier works employing both images and questions. This is primarily due to two factors. First, the question severely constrains the kinds of answers expected in many cases, essentially turning an open-ended question into

**Q:** What are they doing? **A:** Playing baseball
**Q:** What are they playing? **A:** Soccer

**Q:** Is the weather rainy in the picture? **A:** Yes
**Q:** Is it rainy in the picture? **A:** No

Figure 2.13: Slight variations in the way a question is phased causes current VQA algorithms to produce different answers. The left example uses the system in [76] and the right example uses the system from [30].

a multiple-choice one, e.g., questions about the color of an object will have a color as an answer. Second, the datasets tend to have strong bias. These two factors make language a much stronger prior than the image features alone.

The predictive power of language over images have been corroborated by ablation studies. In [125], the authors studied a model that had been trained using both image and question features. They then studied how the predictions of the model differed when it was given only the image or only the question, compared to when it was given both. They found that the image-only model's predictions differed from the combined model 40% more often than the question only model. They also showed that the way the question is phrased strongly biases the answer. When training a neural network, these regularities will be incorporated into the model. While this produces increased performance on the dataset, it is potentially detrimental to creating a general VQA system.

In [14], bias in VQA was studied using synthetic cartoon images. They created a dataset with solely binary questions, in which the same question could be asked about two images that were mostly identical, except for a minor change that caused the correct answer to be different. They found that a model trained on an unbalanced version of this

dataset performed 11% worse (absolute difference) on a balanced test dataset compared to a model trained on a balanced version of the dataset.

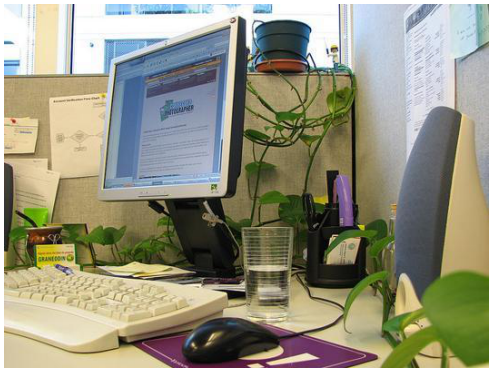
We conducted two experiments to assess the effect of language bias in VQA. First, we used the model[3] from [76]. This model was trained on COCO-VQA, and it allows the contribution of the question and image features to be assessed independently by splitting the weights of the softmax output layer into image and question components. We asked simple binary questions with a relatively equal prior for both choices so that the image must be analyzed to answer the question. Examples are shown in Figure 2.14. We can see that the system performs poorly, especially when considering that the baseline accuracy for yes/no questions for COCO-VQA is about 80%. Next, we studied how language bias affected the more complex MCB-ensemble model [88] that was trained on COCO-VQA. This model was the winner of the 2016 VQA Challenge workshop. To do this, we created a small dataset consisting only of yes/no questions. To create this dataset, we used annotations from the validation split of the COCO dataset to determine whether an image contained a person, and then asked an equal number of 'yes' and 'no' questions about whether there are any people present. We used the questions 'Are there any people in the photo?', 'Is there a person in the picture?', and 'Is there a person in the photo?' For each variation, there were 38,514 yes/no questions (115,542 total). The accuracy of MCB-ensemble on this dataset was worse than chance (47%), which starkly contrasts with its results on COCO-VQA (i.e., 83% on COCO-VQA yes/no questions). This is likely due to severe bias in the training dataset, and not due to an inability for MCB to learn the task.

As shown in Figure 2.13, VQA systems are sensitive to the way a question is phrased. We observed similar results when using the system in [12]. To quantify this issue, we created another toy dataset from the validation split of the COCO dataset and used it to evaluate the MCB-ensemble model that was trained on COCO-VQA. In this toy dataset, the task is to identify which sport was being played. We asked three variations of the same question: 1) 'What are they doing?', 2) 'What are they playing?', and 3) 'What sport are they playing?' Each variation contains 5,237 questions about seven common sports (15,711 questions total). MCB-ensemble achieved 33.6% for variation 1, 78% for variation 2, and 86.4% for variation 3. The dramatic increase in performance from variation 1 to 2 is caused by the inclusion of keyword 'playing' instead of the generic verb 'doing.' The increment from variation 2 to 3 is caused by explicitly including the keyword 'sport' in the question. This suggests that VQA systems are over-dependent on language 'clues' that annotators often include. Taken together, these experiments show that language bias is an issue that critically affects the performance of current VQA systems.

In conclusion, current VQA systems are more dependent on the question than the image content. Language bias in datasets critically affects the performance of the current VQA systems, which limits their deployment. New VQA datasets must endeavor to

[3]An online demo is available here: `http://visualqa.csail.mit.edu/`

**no** (11.07 w/ 2.57 [image] + 8.50 [word])
**yes** (10.94 w/ 2.71 [image] + 8.23 [word])



**yes** (12.45 w/ 4.22 [image] + 8.23 [word])
**no** (12.05 w/ 3.55 [image] + 8.50 [word])

**no** (12.04 w/ 3.54 [image] + 8.50 [word])
**yes** (11.96 w/ 3.72 [image] + 8.23 [word])

**yes** (12.30 w/ 4.07 [image] + 8.23 [word])
**no** (12.14 w/ 3.64 [image] + 8.50 [word])

Figure 2.14: Using the system in [76], the answer score for the question 'Are there any people in the picture?' is roughly the same for 'yes' (8.23) and 'no' (8.50). Answering the question correctly requires examining the image, but the model fails to appropriately use the image information.

compensate for this issue, by either having questions that force analysis of image content and/or by making datasets less biased.

### 2.6.2 How useful is attention for VQA?

It is difficult to determine how much attention helps VQA algorithms. In ablation studies, when attentive mechanisms are removed from models it impairs their performance [88, 96]. Currently, the best model for COCO-VQA does employ spatial visual attention [88], but simple models that do not use attention have been shown to exceed earlier models that used complex attentive mechanisms. In [109], for example, an attention-free model that used multiple global image feature representations (VGG-19, ResNet-101, and ResNet-

152), instead of a single CNN, performed very well compared some attentive models. They combined image and question features using both element-wise multiplication and addition, instead of solely concatenating them. Combined with ensembling, this yielded results significantly higher than the complex attention-based models used in [91] and [94]. Similar results have been obtained by other systems that do not employ spatial attention, e.g, [30, 95, 112]. Attention alone does not ensure good VQA performance, but incorporating attention into a VQA model appears to improve performance over the same model when attention is not used.

In [126], the authors showed that methods commonly used to incorporate spatial attention to specific image features do not cause models to attend to the same regions as humans tasked with VQA. They made this observation using both the attentive mechanisms used in [91] and [96]. This may be because the regions the model learns to attend to are discriminative due to biases in the dataset and not due to where the algorithm should attend. For example, when asked a question about whether drapes are in an image, the algorithm may instead look at the bottom of the image for a bed rather than windows because questions about drapes tend to be found in bedrooms. This is an indication that attentive mechanisms may not be correctly deployed due to biases.

### 2.6.3 Bias Impairs Method Evaluation

Dataset bias significantly impairs the ability to evaluate VQA algorithms. Questions that require the use of the image content are often relatively easy to answer. Many are about the presence of objects or scene attributes. These questions tend to be handled well by CNNs and also have strong language biases. Harder questions, such as those beginning with 'Why' are comparatively rare. This has serious implications for evaluating performance. For COCO-VQA (train and validation partitions), a system that improves accuracy on questions beginning with 'Is' and 'Are' by 15% will increase overall accuracy by 5%. However, the same increase in both 'Why' and 'Where' questions will only increase accuracy by 0.6%. In fact, even if all 'Why' and 'Where' questions are answered correctly, the overall increase in accuracy will only be 4.1%. On the other hand, answering 'yes' to all questions beginning with 'Is there' will yield an accuracy of 85.2% on those questions. These problems could be overcome if each *type* of question was evaluated in isolation, and then the mean accuracy across question types was used instead of overall accuracy for benchmarking the algorithms. This approach is similar to the mean per-class accuracy metric used for evaluating object classification algorithms, which was adopted due to bias in the amount of test data available for different object categories.

### 2.6.4 Are Binary Questions Sufficient?

Using binary (yes/no or true/false) questions to evaluate algorithms has attracted significant discussion in the VQA community. The main argument against using binary questions is the lack of complex questions and the relative ease in answering the questions that are typically generated by human annotators. Visual Genome and Visual7W exclude binary questions altogether. The authors argued that this choice would encourage more complex questions from the annotators.

On the other hand, binary questions are easy to evaluate and these questions can, in theory, encompass an enormous variety of tasks. The SHAPES dataset [82] uses binary questions exclusively but contains complex questions involving spatial reasoning, counting, and drawing inferences (see Figure 2.7). Using cartoon images, [14] also showed that these questions can be especially difficult for VQA algorithms when the dataset is balanced. However, there are challenges to creating balanced binary questions for real world imagery. In COCO-VQA, 'yes' is a much more common answer than 'no,' simply because people tend to ask questions biased toward 'yes' as an answer.

As long as bias is controlled, yes/no questions can play an important role in future VQA benchmarks, but a VQA system should be capable of more than solely binary questions so that its abilities can be fully assessed. All real-world applications for VQA, such as enabling the blind to ask questions about visual content, require the output of the VQA system to be open-ended. A system that can solely handle binary questions will have limited real-world utility.

### 2.6.5 Open Ended vs. Multiple Choice

Because it is challenging to evaluate open-ended multi-word answers, multiple-choice has been proposed as a way to evaluate VQA algorithms. As long as the alternatives are sufficiently difficult, a system could be evaluated in this manner but then be deployed to answer open-ended questions. For these reasons, multiple choice is used to evaluate Visual7W, Visual Genome, and a variant of The VQA Dataset. In this framework, an algorithm has access to a number of possible answers (e.g., 18 for COCO-VQA), along with the question and image. It must then select among possible choices.

A major problem with multiple-choice evaluation is that the problem can be reduced to determining which of the answers is correct instead of actually answering the question. For example, in [112], they formulated VQA as an answer scoring task, where the system was trained to produce a score based on the image, question, and potential answers. The answers themselves were fed into the system as features. It achieved state-of-the-art results on Visual7W and rivals the best methods on COCO-VQA, with their method performing better than many complex systems that use attention. To a large extent, we believe their system performed well because it learned to better exploit biases in the an-

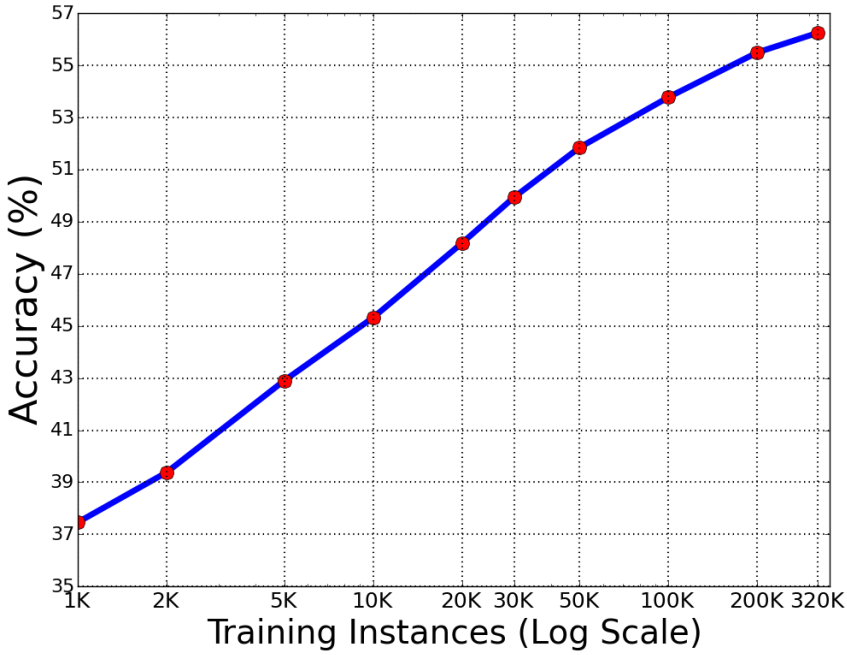



Figure 2.15: The graph showing test accuracy as a function of available training data on the COCO-VQA dataset.

swers instead of reasoning about images. On Visual7W, they showed that a variant of their system that used solely the answers and was both image- and question-blind rivaled baselines using the question and image.

We argue that any VQA system should be able to operate without being given answers as inputs. Multiple-choice can be an important ingredient for evaluating multi-word answers, but it alone is not sufficient. When multiple-choice is used, the choices must be selected carefully to ensure that a question is hard and not deducible from the provided answers alone. A system that is solely capable of operating with answers provided is not really solving VQA, because these are not available when a system is deployed.

## 2.7 Recommendations for Future VQA Datasets

Existing VQA benchmarks are not sufficient to evaluate whether an algorithm has 'solved' VQA. In this section, we discuss future developments in VQA datasets that will make them better benchmarks for the problem.

Future datasets need to be larger. While VQA datasets have been growing in size and diversity, algorithms do not have enough data for training and evaluation. We did a small experiment where we trained a simple MLP baseline model for VQA using ResNet-152

image features and skip-thought features for the questions, and we assessed performance as a function of the amount of training data available on COCO-VQA. The results are shown in Figure 2.15, where it is clear that the curve has not started to approach an asymptote. This suggests that even on datasets that are biased, increasing the size of the dataset could significantly improve accuracy. However, this does not mean that increasing the size of the dataset is sufficient to turn it into a good benchmark, because humans tend to create questions with strong biases.

Future datasets need to be less biased. We have repeatedly discussed the problem of bias in existing VQA datasets in this chapter, and pointed out the kinds of problems these biases cause for truly evaluating a VQA algorithm. For real-world open-ended VQA, this will be difficult to achieve without carefully instructing the humans that generate the questions. Bias has long been a problem in images used for computer vision datasets (for a review see [127]), and for VQA this problem is compounded by bias in the questions as well.

In addition to being larger and less biased, future datasets need more nuanced analysis for benchmarking. All of the publicly released datasets use evaluation metrics that treat every question with equal weight, but some kinds of questions are far easier, either because of bias or because existing algorithms excel at answering that kind of question, e.g., object recognition questions. Some datasets such as COCO-QA have divided VQA questions into distinct categories, e.g., for COCO-QA these are color, counting (number), location, and object. We believe that mean per-question type performance should replace standard accuracy, so each question would not have equal weight in evaluating performance. This would go a long way towards making a VQA algorithm have to perform well at a wide variety of question types to perform well overall, otherwise a system that excelled at answering 'Why' questions but was slightly worse than another model at more common questions would not be fairly evaluated. To do this, each question would need to be assigned a category. We believe this effort would make benchmark results significantly more meaningful. The scores on each question type could also be used to compare algorithms to see which kind of questions they excel at. We explore creation of such a dataset in Chapter 4.

## 2.8 Conclusions

VQA is an important basic research problem in computer vision and natural language processing that requires a system to do much more than task specific algorithms, such as object recognition and object detection. An algorithm that can answer arbitrary questions about images would be a milestone in artificial intelligence. We believe that VQA should be a necessary part of any visual Turing test.

In this chapter, we critically reviewed existing datasets and algorithms for VQA. We

discussed the challenges of evaluating answers generated by algorithms, especially multi-word answers. We described how biases and other problems plague existing datasets. This is a major problem, and the field needs a dataset that evaluates the important characteristics of a VQA algorithm, so that if an algorithm performs well on that dataset then it means it is doing well on VQA in general.

Future work on VQA involves the creation of larger and far more varied datasets. Bias in these datasets will be difficult to overcome, but evaluating different kinds of questions individually in a nuanced manner, rather than using naive accuracy alone, will help significantly. Further work will be needed to develop VQA algorithms that can reason about image content, but these algorithms may lead to significant new areas of research.

Table 2.2: Comparison of different evaluation metrics proposed for VQA.

| | Pros | Cons |
|---|---|---|
| **Simple Accuracy** | • Very simple to evaluate and interpret<br>• Works well for small number of unique answers | • Both minor and major errors are penalized equally<br>• Can lead to explosion in number of unique answers, especially with presence of phrasal or sentence answers |
| **Modified WUPS** | • More forgiving to simple variations and errors<br>• Does not require exact match<br>• Easy to evaluate with simple script | • Generates high scores for answers that are lexically related but have diametrically opposite meaning<br>• Cannot be used for phrasal or sentence answers |
| **Consensus Metric** | • Common variances of same answer could be captured<br>• Easy to evaluate after collecting consensus data | • Can allow for some questions having two correct answers<br>• Expensive to collect ground truth<br>• Difficulty due to lack of consensus |
| **Manual Evaluation** | • Variances to same answer is easily captured<br>• Can work equally well for single word as well as phrase or sentence answers | • Can introduce subjective opinion of individual annotators<br>• Very expensive to setup and slow to evaluate, especially for larger datasets |

Table 2.3: Results across VQA datasets for both open-ended (OE) and multiple-choice (MC) evaluation schemes. Simple models trained only on the image data (IMG-ONLY) and only on the question data (QUES-ONLY) as well as human performance are also shown. IMG-ONLY and QUES-ONLY models are evaluated on the 'test-dev' section of COCO-VQA. MCB-ensemble [88] and AMA [75] are presented separately as they use additional data for training.

| | **DAQUAR** | | **COCO-QA** | **COCO-VQA** | |
|---|---|---|---|---|---|
| | **FULL** | **37** | | **OE** | **MC** |
| IMG-ONLY [30] | 6.19 | 7.93 | 34.36 | 29.59 | - |
| QUES-ONLY [30] | 25.57 | 39.66 | 39.24 | 49.56 | - |
| MULTI-WORLD [25] | 7.86 | 12.73 | - | - | - |
| ASK-NEURON [78] | 21.67 | 34.68 | - | - | - |
| ENSEMBLE [71] | - | 36.94 | 57.84 | - | - |
| LSTM Q+I [12] | - | - | - | 54.06 | 57.17 |
| iBOWIMG [76] | - | - | - | 55.89 | 61.97 |
| DPPNet [89] | 28.98 | 44.48 | 61.19 | 57.36 | 62.69 |
| SMem [90] | - | 40.07 | - | 58.24 | - |
| SAN [91] | 29.3 | 45.5 | 61.6 | 58.9 | - |
| NMN [82] | - | - | - | 58.7 | - |
| D-NMN [92] | - | - | - | 59.4 | - |
| FDA [93] | - | - | - | 59.54 | 64.18 |
| HYBRID [30] | 28.96 | 45.17 | 63.18 | 60.06 | - |
| DMN+ [94] | - | - | - | 60.4 | - |
| MRN [95] | - | - | - | 61.84 | 66.33 |
| HieCoAtten [96] | - | - | 65.4 | 62.1 | 66.1 |
| RAU_ResNet [97] | - | - | - | 63.2 | 67.3 |
| DAN [98] | - | - | - | 64.2 | 69.0 |
| MCB+Att [88] | - | - | - | 64.2 | - |
| MLB [99] | - | - | - | 65.07 | 68.89 |
| AMA [75] | - | - | 69.73 | 59.44 | - |
| MCB-ensemble [88] | - | - | - | 66.5 | 70.1 |
| **HUMAN** | **50.20** | **60.27** | **-** | **83.30** | **91.54** |

Table 2.4: Overview of different methods that were evaluated on open-ended COCO-VQA and their design choices. Results are report on the 'test-dev' split when 'test-standard' results are not available (Denoted by *).

| Method | Accuracy (%) ($Acc_{VQA}$) | CNN Network | Use of Attention | Ext. Data | Compo-sitional |
|---|---|---|---|---|---|
| LSTM Q+I [12] | 54.1 | VGGNet | - | - | - |
| iBOWIMG [76] | 55.9 | GoogLeNet | - | - | - |
| DPPNet [89] | 57.4 | VGGNet | - | - | - |
| SMem [90] | 58.2 | GoogLeNet | ✓ | - | - |
| SAN [91] | 58.9 | GoogLeNet | ✓ | - | - |
| NMN [82] | 58.7 | VGGNet | ✓ | - | ✓ |
| D-NMN [92] | 59.4 | VGGNet | ✓ | - | ✓ |
| AMA [75] | 59.4 | VGGNet | - | ✓ | - |
| FDA [93] | 59.5 | ResNet | ✓ | - | |
| HYBRID [30] | 60.1 | ResNet | - | - | - |
| DMN+ [94] | 60.4 | ResNet | ✓ | - | - |
| MRN [95] | 61.8 | ResNet | ✓ | - | - |
| HieCoAtten-VGG* [96] | 60.5 | VGGNet | ✓ | - | - |
| HieCoAtten-ResNet [96] | 62.1 | ResNet | ✓ | - | - |
| RAU_VGG* [97] | 61.3 | VGGNet | ✓ | - | - |
| RAU_ResNet [97] | 63.2 | ResNet | ✓ | - | |
| MCB* [88] | 61.2 | ResNet | - | - | - |
| MCB-ATT* [88] | 64.2 | ResNet | ✓ | - | - |
| DAN-VGG* [98] | 62.0 | VGGNet | ✓ | - | - |
| DAN-ResNet [98] | 64.3 | ResNet | ✓ | - | - |
| MLB [99] | 65.1 | ResNet | ✓ | - | - |
| MLB+VG* [99] | 65.8 | ResNet | ✓ | ✓ | - |
| MCB-ensemble [88] | 66.5 | ResNet | ✓ | ✓ | - |

# Chapter 3

# Answer-Type Prediction for Visual Question Answering

## 3.1 Introduction

A natural way to address combined vision and language understanding is open-ended visual question answering (VQA) which we described in Chapter 2. VQA is especially challenging because models for VQA need to be implicitly capable of object recognition, object detection, attribute recognition, and more. In this chapter, we describe a novel algorithm for VQA incorporates Bayesian framework and combine with a discriminative model to achieve better results on four different datasets compared to prior algorithms.

Existing algorithms treat each question equally in a black-box setup and do not explicitly leverage the fact that question alone can provide important clues about the possible answer. Our main contribution is to observe that when answering a question, it is generally possible to predict the form the answer will take. For example, for the question "Is it raining?" a valid answer will be either "yes" or "no." The answer will never be "green" or "10." However, existing models do not have this kind of reasoning explicitly built into them. Incorporating information predicted about the answer can also potentially improve a model's internal representation to handle the question.

We first describe a Bayesian framework for VQA that incorporates answer-type prediction. We then show that we can use text-based features to predict with greater than 99% accuracy the form the answer will take for all of the datasets. We then evaluate and compare our model against methods from the literature and a discriminative model trained using the same features. Another contribution is the use of skip-thought vectors [114], which have not previously been used for VQA. Skip-thought vectors are a recently developed technique for encoding sentences into vectors in a manner that preserves salient sentence information. We are also the first to evaluate our models on each of the publicly available datasets for VQA, and we provide a critical analysis of each dataset's strengths

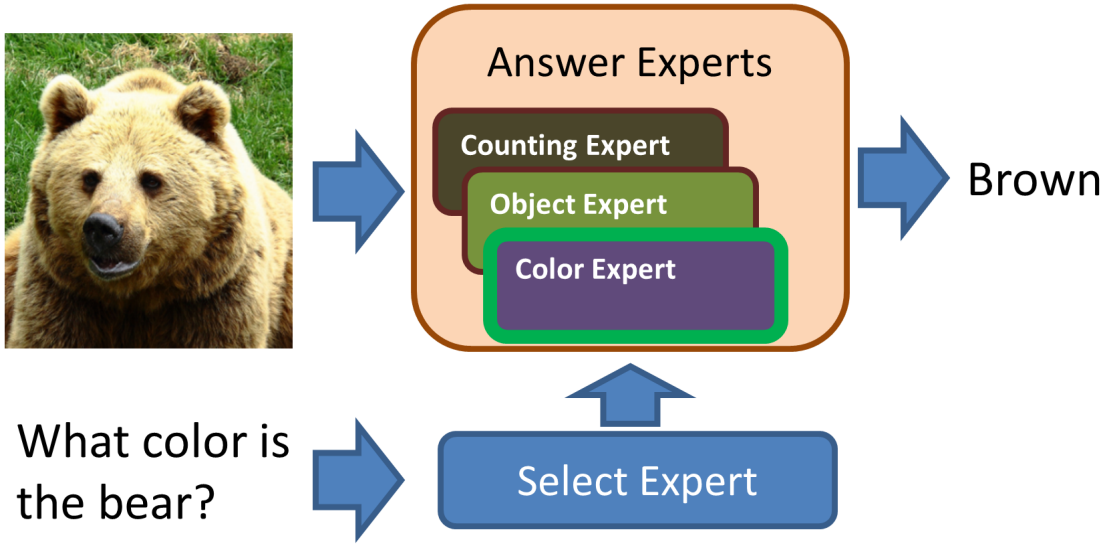



Figure 3.1: In the open-ended VQA problem, an algorithm is given an image and a question, and it must output a string containing the answer. We obtain state-of-the-art results on multiple VQA datasets by adopting a Bayesian approach that incorporates information about the form the answer should take. In this example, the system is given an image of a bear and it is asked about the color of the bear. Our method explicitly infers that this is a "color" question and uses that information in its predictive process.

and weaknesses. A demonstration of a simplified version of our algorithm can be found at `http://askimage.org`.

## 3.2 Related Work

We outlined several algorithms for VQA in Chapter 2. Here we will briefly describe VQA algorithms that we use as comparison to our model.

In [12], the authors' best model on COCO-VQA was an LSTM model with a 1000-node softmax output layer, which generated answers for the top-1000 most frequent answers. Their LSTM model used a one-hot encoding of question words and CNN features from a pre-trained network. A linear transformation mapped the CNN features to the same dimensionality of the question words. These were then combined using the element-wise (Hadamard) product, and then fed into a MLP network.

In [71], a similar approach was taken, with the main difference being that they fed CNN features to the LSTM as the first "word," followed by vectors encoding each word of the sentence, and then finally the last word was the CNN features once more. In a

DAQUAR: What colors do the stools around the table have ?
Ground Truth: blue, white
DAQUAR: What is leaning against the wall on the left side of the white cabinet?
Ground Truth: Ladder

COCO-VQA: Do the horses legs look strong enough to support its body?
Ground Truth: Yes
COCO-QA: What is the color of the horses?
Ground Truth: Brown

COCO-VQA: What direction are the giraffes facing?
Ground Truth: to right; away;left; 3 towards camera; away from camera; forward; west; toward building
COCO-QA: What just standing around an old stable as their picture is taken?
Ground Truth: Giraffes

Figure 3.2: Images and their corresponding question-answer pairs from the COCO-VQA, COCO-QA, and DAQUAR datasets. DAQUAR is generated using human annotators and it has unambiguous questions, but it has solely indoor images that tend to have many small objects. COCO-VQA is generated by human annotators and has a wide variety of questions, but some questions have ambiguous or subjective answers. COCO-QA is generated using an automated algorithm that produces one word answers, but some questions are grammatically incorrect.

variant of this approach, [78] sequentially gave their LSTM network concatenated CNN and word features at every time step.

In [72], separate LSTMs were used for the question and answer, but they had a shared word embedding layer, and CNN image features are fused at the end of the LSTM. Their model was able to output more than one-word answers or lists, and it could generate coherent sentences.

As an alternative to LSTM networks, in [25] the authors created a Bayesian framework for VQA. They used semantic segmentation to get information about the objects present in an image, such as their categories and spatial locations. Then, their Bayesian framework calculated the probability of each answer given the semantic segmentation image features and the question.

Unlike us, none of these methods explicitly incorporated information about the answer-type. Also, instead of using an LSTM network, we encode questions using skip-thought vectors (see section 3.5.4).

## 3.3 Evaluation of VQA Systems

The most straightforward measure used to evaluate VQA systems is accuracy, *i.e.* , the system must output exactly the same answer as the human annotator. However, difficulties can arise due to this. First, many questions have multiple valid answers, *e.g.* , "What is on the table?" might have "Mandarin Orange," "orange," or "fruit" as valid answers. Using one-to-one matching will penalize a model if it does not output exactly the same answer as the human annotator. Several alternatives have been proposed, which we described in chapter 2. In chapter 2, we also discussed several problems with each of the metrics. Despite those problems, we used the standard evaluation metrics as described by the authors for each dataset in order to allow direct comparison with existing results.

For DAQAUR dataset, we use standard accuracy and WUPS metric as originally described by the authors.For COCO-VQA dataset, the standard VQA accuracy metric from [12] is used. As a reminder, it is defined as:

$$\text{Accuracy}_{\text{VQA}} = \min(\frac{n}{3}, 1), \tag{3.1}$$

where $n$ is the number of people that gave the predicted answer.

## 3.4 Predicting the Answer-Type

Our method requires each question to be assigned a type during training. The way we do this differs for each dataset.

DAQUAR does not have explicitly defined answer categories. We created three categories by looking at the answers: Number, Color, and Other. We assigned all answers that were numbers to the number category, all answers that were one of the 10 canonical colors (black, white, blue, brown, gray, green, orange, purple, red) to the color category, all other answers were assigned to the other category.

COCO-QA has four explicitly defined answer categories: Object, Color, Counting (Number), and Location. We did not change them. For COCO-VQA, 'Yes/No', 'Number', and 'Other' types are explicitly defined (denoted DT for default types). Besides using the default types, we also used an extended set of types that we constructed with heuristics (denoted ET for extended types). We subdivided the 'Number' category into 'counting' and 'other numbers' by looking at whether the question began with 'how many.' Answers that were 14 common colors (black, white, blue, brown, gray, green, orange, purple, red, silver, gold, tan and pink) were assigned to the color category, if the question contained the word 'color'. 'COCO objects' was assigned if the answer was one of the object categories defined in COCO. Finally, all questions terminating in 'playing' or 'doing' were assigned the type 'activity.' All remaining questions were grouped under the 'others' type category.

Across all of the datasets, we were able to use our skip-thought representation with logistic regression to infer the answer type for questions with over 99.7% accuracy on validation data.

## 3.5 Models for VQA

### 3.5.1 A New Bayesian Model for VQA

We formulate the VQA problem in a Bayesian framework. Let $\mathbf{x}$ be a column vector containing image features and $\mathbf{q}$ be a column vector containing question features. Given a question and an image, our model estimates the probability of a particular answer $k$ and question-type $c$ as $P(A = k, T = c|\mathbf{x}, \mathbf{q})$. Using Bayes' rule and the chain rule for probabilities, this can be expressed as

$$P(A = k, T = c|\mathbf{x}, \mathbf{q}) = \frac{P(\mathbf{x}|A = k, T = c, \mathbf{q})\, P(A = k|T = c, \mathbf{q})\, P(T = c|\mathbf{q})}{P(\mathbf{x}|\mathbf{q})},$$

where $P(\mathbf{x}|A = k, T = c, \mathbf{q})$ is the probability of the image features given the answer, answer-type, and question, $P(A = k|T = c, \mathbf{q})$ is the probability of the answer given the answer-type, and question, $P(T = c|\mathbf{q})$ is the probability of the answer-type given the question, and $P(\mathbf{x}|\mathbf{q})$ is the probability of the image features given the question. To obtain the answer to a question about an image, we can simply marginalize over all of the answer types, *i.e.* ,

$$P(A = k|\mathbf{x}, \mathbf{q}) = \sum_{c \in T} P(A = k, T = c|\mathbf{x}, \mathbf{q}).$$

While it is possible to train all aspects of the model jointly using a maximum likelihood solution, we chose to use simple models that are trained individually for each distribution. This makes training simple and fast. We model $P(A = k|T = c, \mathbf{q})$ and $P(T = c|\mathbf{q})$ using logistic regression classifiers. Because $P(\mathbf{x}|\mathbf{q})$ does not influence the prediction, it can be disregarded.

We model each $P(\mathbf{x}|A = k, T = c, \mathbf{q})$ with a conditional multivariate Gaussian, *i.e.* ,

$$P(\mathbf{x}|A = k, T = c, \mathbf{q}) = \mathcal{N}\left(\mathbf{x}|\bar{\boldsymbol{\mu}}_{k,c,\boldsymbol{q}}, \bar{\boldsymbol{\Sigma}}_{k,c}\right).$$

This approach shares similarities with attention, in that it directly models that the image features that should be paid attention to should depend on the question. It is related to Quadratic Discriminant Analysis (QDA) [128]; however, in standard QDA the Gaussians are not conditional on additional features, unlike our approach.

The conditional mean and covariance for each Gaussian is computed as follows. Let the sample mean and covariance for the training data with answer $k$ and answer-type $c$,

in which the image features $\boldsymbol{x}$ are concatenated with the question features $\boldsymbol{q}$, be $\boldsymbol{\mu}_{k,c} = \begin{bmatrix} \boldsymbol{\mu}_{k,c,\mathbf{x}} & \boldsymbol{\mu}_{k,c,\mathbf{q}} \end{bmatrix}^T$ and

$$\boldsymbol{\Sigma}_{k,c} = \begin{bmatrix} \boldsymbol{\Sigma}_{k,c,1,1} & \boldsymbol{\Sigma}_{k,c,1,2} \\ \boldsymbol{\Sigma}_{k,c,2,1} & \boldsymbol{\Sigma}_{k,c,2,2} \end{bmatrix}.$$

Then, the mean of the Gaussian given $\boldsymbol{q}$ is

$$\bar{\boldsymbol{\mu}}_{k,c,\mathbf{q}} = \boldsymbol{\mu}_{k,c,\mathbf{x}} + \boldsymbol{\Sigma}_{k,c,1,2}\boldsymbol{\Sigma}_{k,c,2,2}^{-1}\left(\mathbf{q} - \boldsymbol{\mu}_{k,c,\mathbf{q}}\right)$$

and the covariance will be

$$\bar{\boldsymbol{\Sigma}}_{k,c} = \boldsymbol{\Sigma}_{k,c,1,1} - \boldsymbol{\Sigma}_{k,c,1,2}\boldsymbol{\Sigma}_{k,c,2,2}^{-1}\boldsymbol{\Sigma}_{k,c,2,1}.$$

Note that the new mean for the image features depends on the question features, but the new covariance does not.

Because we have limited training data for some answer and answer-type combinations, estimating $\boldsymbol{\Sigma}_{k,c}$ accurately is difficult and the estimate should be regularized to ensure we can invert the covariance sub-matrices. To remedy this, we use a locally smoothed solution combined with shrinkage to estimate $\boldsymbol{\Sigma}_{k,c}$ [129], which is given by

$$\boldsymbol{\Sigma}_{k,c} = \frac{n_{k,c}\left(1-\beta\right)\boldsymbol{\Sigma}'_{k,c} + \frac{1}{\kappa}\beta \sum\limits_{j \in KNN(k,c)} n_{j,c}\boldsymbol{\Sigma}'_{j,c}}{n_{k,c}\left(1-\beta\right) + \frac{1}{\kappa}\beta \sum\limits_{j \in KNN(k,c)} n_{j,c}} + \epsilon\mathbf{I}$$

where $\boldsymbol{\Sigma}'_{k,c}$ is the sample covariance matrix for the data with answer $k$ and answer-type $c$ and $n_{k,c}$ is the corresponding number of samples, $\mathbf{I}$ is the identity matrix, $\epsilon$ and $\beta$ are scalar regularization parameters, and KNN$(\cdot)$ denotes the categories of the same type that have means with the smallest $\kappa$ Euclidean distances to $\boldsymbol{\mu}_{k,c}$. We used $\kappa = 10$, $\epsilon = 0.01$, and $\beta = 0.4$ in all of our experiments.

In preliminary experiments, we also tried modeling $P\left(\mathbf{x}|A = k, T = c, \mathbf{q}\right)$ using conditionalized kernel density estimation with Gaussian kernels and using a conditionalized Gaussian mixture model, but in both cases performance was significantly worse on validation data than simply using a single Gaussian per answer.

### 3.5.2 Baseline Models

In addition to comparing to the models in the literature, we also tested five baseline models ourselves.

1. IMAGE: A logistic regression classifier trained with image features. It knows nothing about the question.

2. IMAGE+TYPE: For each answer-type in the dataset, we train a logistic regression classifier. We use our answer-type prediction model to select among the logistic regression classifiers for a given question, but the classifier does not have access to detailed question information. A similar approach was used in [71], where they used a question-type oracle to select among image feature classifiers on COCO-QA.

3. QUESTION: A logistic regression classifier trained only with the question features.

4. IMAGE+QUESTION: A logistic regression classifier trained with the image features concatenated to the question features.

5. MLP: A multi-layer perceptron network with a softmax output layer, with the image and question features as input. MLP is a 4-layer neural network with 6000 units on the first layer, 4000 for the second, 2000 for the third, and finally a softmax output layer with units equal to the number of categories. All hidden layers used rectified linear units. To regularize the network, drop-out of 0.3 was used with the hidden layers as well the input data layer.

### 3.5.3 Hybrid Model

Hybrid models that combine generative and discriminative classifiers can achieve a lower error rate than either alone [130]. Motivated by this, we created a hybrid approach that multiplicatively combines the two models, *i.e.* ,

$$P_H\left(A=k|\mathbf{x},\mathbf{q}\right) \propto P_B\left(A=k|\mathbf{x},\mathbf{q}\right) P_D\left(A=k|\mathbf{x},\mathbf{q}\right)^{\alpha},$$

where $P_B\left(A=k|\mathbf{x},\mathbf{q}\right)$ is our Bayesian model, $P_D\left(A=k|\mathbf{x},\mathbf{q}\right)$ is IMAGE+QUESTION as described earlier, and $\alpha$ is a parameter that weights the distributions appropriately. This kind of weighting is a common approach to combining classifiers that were independently trained [131]. For DAQUAR and COCO-QA, we do five-fold cross-validation on the training data to find a good value for $\alpha$, and for COCO-VQA $\alpha$ is tuned using the validation data. In both cases, we searched for $\alpha$ over $0.0, 0.1, 0.2, \ldots, 6$. This approach is labeled HYBRID. Additionally, for COCO-VQA, we used a variation where we combined our Bayesian model with MLP (HYBRID-MLP).

### 3.5.4 Question and Image Feature Representations

We use skip-thought vectors [114] to encode the text of a question into a vector $\boldsymbol{q}$, which have not previously been used for VQA. Skip-thought vectors are trained in an encoder-decoder framework, in which both the encoder and decoder are recurrent neural networks with gated recurrent units. The model is trained to encode a sentence, and it uses that

encoding to reconstruct the previous and next sentence. They can therefore be trained in an unsupervised manner from corpora of text. After training, the output of the encoder can be used as a rich feature vector, which was shown to achieve excellent performance on a variety of NLP classification tasks when used with a linear classifier. We use the 4800-dimensional combine-skip model from [114], which is a concatenation of uni-skip and bi-skip models. Each skip-thought vector is normalized to unit length.

For our image features $\boldsymbol{x}$, we used ResNet [4], with $448 \times 448 \times 3$ images. The features were taken from the last hidden layer after the ReLU, and then pooled across all spatial locations. These features were normalized to unit length. For the Bayesian model, we reduced the dimensionality of the CNN features using linear discriminant analysis to $K - 1$ dimensions, where $K$ is the number of possible answers.

## 3.6 Experiments

### 3.6.1 DAQUAR

Results on DAQUAR are shown in Table 3.1. For accuracy, both DAQUAR-FULL, QUESTION, IMAGE+QUESTION, BAYESIAN, and HYBRID all outperformed the prior state-of-the-art, with HYBRID performing best. A similar trend occurred for DAQUAR-37, with the exception of IMAGE+QUESTION. In both cases, we observe that QUESTION alone exceeds the previous state-of-the-art, and it performs extremely well compared to IMAGE alone. On DAQUAR-FULL, QUESTION achieves only slightly lower accuracy than the best performing HYBRID method. This may be because we and others used off-the-shelf CNN features that were tuned to recognize objects on ImageNet. These features tend to do a significant amount of spatial pooling, but DAQUAR questions are often about small objects in the image.

### 3.6.2 COCO-QA

Results on COCO-QA are shown in Table 3.1. For both accuracy and WUPS, HYBRID performed best, even though there was a gap in the performance of BAYESIAN and IMAGE+QUESTION. This suggests that the models are complementary, and they are making different mistakes. This did not occur with DAQUAR, and it may be because we have a lot more training data per answer on average for COCO-QA than we have for DAQUAR. We investigate this further in Section 3.6.5.

Table 3.1: Results on DAQUAR-FULL, DAQUAR-37, and COCO-QA. All results on DAQUAR are for one-word answers.

| | **DAQUAR-FULL** | | | **DAQUAR-37** | | | **COCO-QA** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc. (%) | WUPS 0.9 | WUPS 0.0 | Acc. (%) | WUPS 0.9 | WUPS 0.0 | Acc. (%) | WUPS 0.9 | WUPS 0.0 |
| MULTI-WORLD [25] | 7.86 | 11.86 | 38.79 | 12.73 | 18.10 | 51.47 | - | - | - |
| ASK-NEURON [78] | 21.67 | 27.99 | 65.11 | 34.68 | 40.76 | 79.54 | - | - | - |
| TORONTO-FULL [71] | - | - | - | 36.94 | 48.15 | 82.68 | 57.84 | 67.90 | 89.52 |
| IMAGE | 6.19 | 11.31 | 45.83 | 7.93 | 13.13 | 54.38 | 34.36 | 46.63 | 72.58 |
| QUESTION | 25.57 | 31.49 | 67.09 | 39.66 | 44.19 | 82.19 | 39.24 | 50.11 | 83.42 |
| IMAGE+TYPE | 13.36 | 20.28 | 61.37 | 17.59 | 24.51 | 75.61 | 48.31 | 63.16 | 87.37 |
| IMAGE+QUESTION | 26.83 | 32.86 | 66.86 | 38.28 | 43.83 | 82.45 | 62.27 | 72.36 | 90.99 |
| MLP | 24.05 | 29.96 | 63.61 | 41.72 | 47.00 | 83.27 | 60.84 | 71.03 | 90.65 |
| BAYESIAN | 28.39 | 34.19 | **67.48** | 43.79 | 48.42 | 84.31 | 59.02 | 69.38 | 90.12 |
| HYBRID | **28.96** | **34.74** | 67.33 | **45.17** | **49.74** | **85.13** | **63.18** | **73.14** | **91.32** |

### 3.6.3 COCO-VQA

There are two subsets of COCO-VQA that are used for evaluation: Test-Dev and Test-Standard. The ground truth of both subsets is held by the creators of COCO-VQA, and it is necessary to upload predicted answers to their server for evaluation. Test-Dev is intended for development purposes, and Test-Standard is used to compare state-of-the-art methods. Researchers are currently only allowed to submit five results on Test-Standard and only one result file per day. We benchmark all of our methods on Test-Dev, and we benchmark the best performing of these on Test-Standard.

In our experiments on DAQUAR and COCO-QA, we trained our model to answer all answers in the training data; however, the number of possible answers is far greater for COCO-VQA (see Table 4.1). For COCO-VQA, we selected the most repeated answer for each question, and of these we only used top 1000 most common answers. This covers 82.67% of the answers in train and validation sets [12]. We did not use the remaining training data. Our results on COCO-VQA are given in Table 3.2. HYBRID methods performed well, with HYBRID-MLP using extended answer-types performing best on Test-Dev.

### 3.6.4 Visual7W

Visual7W results are shown in Table 3.3. Following [73], we show both top-1 accuracy and top-5 accuracy. For our experiments, the model was trained only with answers that occurred at least 20 times (536 total categories). HYBRID worked best, and it exceeded the prior state-of-the-art [73].

### 3.6.5 Bayesian vs. Discriminative

Generative models have been reported to outperform discriminative models when the amount of training data is low [132]. To investigate if this was the case here, we studied the difference in performance between our BAYESIAN and IMAGE+QUESTION models on answers with a different number of training examples on COCO-QA. We computed the median and mean number of examples for the training answers in which the Bayesian model performed better than the discriminative model and vice versa. For the answers in which the Bayesian model performed better, the median was 66 and the mean was 163.7, and for discriminative the median was 90 and the mean was 298.9. This is consistent with earlier observations [132], although it is somewhat surprising because the covariance matrices used in our Bayesian model require a significant amount of data to accurately estimate.

Table 3.2: Results on COCO-VQA, which were computed by uploading our models' predictions to the server run by the dataset's creators [12]. We compare against the best results in [12]. Key: IMG=IMAGE, QUES=QUESTION, BAYES=BAYESIAN, HYB=HYBRID, QTYPE=QUESTION TYPE DT=Default types, ET=Extended types.

| | All | Yes/No | Number | Other |
|---|---|---|---|---|
| | **Test Development** | | | |
| LSTM Q+I [12] | 53.74 | 78.94 | 35.24 | 36.42 |
| IMG | 29.59 | 70.65 | 0.38 | 1.16 |
| QUES | 49.56 | 77.36 | 35.49 | 29.02 |
| IMG+QTYPE-DT | 36.02 | 69.53 | 36.03 | 7.44 |
| IMG+QTYPE-ET | 44.74 | 69.49 | 34.76 | 25.88 |
| IMG+QUES | 54.92 | 76.92 | 35.77 | 40.46 |
| BAYES-DT | 53.49 | 77.00 | 35.13 | 37.57 |
| BAYES-ET | 54.58 | 77.58 | 35.03 | 39.66 |
| HYB-DT | 55.68 | 77.25 | 36.29 | 41.65 |
| HYB-ET | 56.00 | 77.21 | 36.10 | 42.38 |
| MLP | 58.65 | 79.93 | 36.80 | 45.42 |
| HYB-MLP-DT | 59.30 | 80.26 | 37.03 | 46.37 |
| HYB-MLP-ET | **59.57** | **80.47** | **37.50** | **46.72** |
| | **Test Standard** | | | |
| LSTM Q+I [12] | 54.06 | 79.01 | 35.55 | 36.80 |
| HYB-MLP-ET | **60.06** | **80.34** | **37.82** | **47.56** |

COCO-VQA: What are they playing?
Ground Truth: N/A Predicted: Frisbee

COCO-VQA: What kind of lens is used in this photo?
Ground Truth: N/A Predicted: Fire Hydrant

COCO-QA: What does the small child eat at the table?
Ground Truth: Donut Predicted: Donut

COCO-QA: What does the red , two level bus with it ; s open on the street and passengers inside the bus?
Ground Truth: Doors Predicted: Bus

DAQUAR: What is on the left side of the fire extinguisher and on the right side of the chair?
Ground Truth: Table Predicted: Table

Visual7W: 2. What color is the sidewalk
Ground Truth: Gray Predicted: Gray
Visual7W: 1. Where are the men talking?
GT: Sidewalk Predicted: In the street

Figure 3.3: Examples of correctly and incorrectly answered questions from each dataset using HYBRID for all datasets, except COCO-VQA where HYBRID-MLP-ET is shown. Because we do not have the answers for COCO-VQA's test datasets, we chose examples that subjectively looked correct or wrong to us.

### 3.6.6 Does Answer-Type Prediction Help Accuracy?

Our proposed model directly incorporates answer-type prediction, but how useful is it? We studied this on DAQUAR-FULL, DAQUAR-37, and COCO-QA by doing experiments with a variant of our Bayesian model that did not incorporate explicit answer-type prediction. For DAQUAR-FULL and DAQUAR-37, we achieved similar performance with and without answer-type prediction (less than 0.5% difference in both cases). However, for COCO-QA we did find a meaningful improvement in performance with answer-type prediction, improving accuracy from 57.33% without answer-types to 59.02%. We also found that expanding the number of answer types improved accuracy on COCO-VQA.

Table 3.3: Top-1 accuracy and top-5 accuracy on Visual7W.

| | **Acc. (%)** | **Top-5 Acc. (%)** |
|---|---|---|
| LSTM (Q+I) [73] | 18.8 | 41.3 |
| IMAGE | 3.76 | 12.25 |
| QUESTION | 17.17 | 36.90 |
| IMAGE+TYPE | 8.39 | 25.70 |
| IMAGE+QUESTION | 22.07 | 43.34 |
| MLP | 20.76 | 42.73 |
| BAYESIAN | 19.23 | 39.83 |
| HYBRID | **22.29** | **43.58** |

## 3.7 Conclusion

In this chapter, we proposed a Bayesian model for VQA that incorporates answer-type prediction, and we found that when it was combined with a discriminative model it achieved excellent results on four VQA datasets. Our Bayesian model is related to QDA, but we modified it to have a visual feature representation that is conditioned on the question features. We pioneered the use of skip-thought vectors for VQA, and we critically reviewed evaluation measures and datasets for open-ended VQA.

# Chapter 4

# An Analysis of Visual Question Answering Algorithms

## 4.1 Introduction

As we discussed in Chapter 2, VQA research began in earnest when several datasets started being available and the progress has been swift. In chapter 3, we described our algorithm which improved upon prior algorithms. However, several new algorithms have been proposed that far surpass our algorithm. On the most popular dataset, 'The VQA Dataset' [12], the best algorithms are now around 70% accuracy [88] (human performance is 83%). While these results are promising, there are critical problems with existing datasets in terms of multiple kinds of biases. Moreover, because existing datasets do not group instances into meaningful categories, it is not easy to compare the abilities of individual algorithms. For example, one method may excel at color questions compared to answering questions requiring spatial reasoning. Because color questions are far more common in the dataset, an algorithm that performs well at spatial reasoning will not be appropriately rewarded for that feat due to the evaluation metrics that are used. We briefly discussed this problem in chapter 2, where we envisioned a better formulated dataset. In this chapter, we discuss such a dataset and discuss how several newly introduced algorithms fare when subjected to a more nuanced analysis.

**Contributions**: This chapter provides four major contributions aimed at better analyzing and comparing VQA algorithms: 1) We create a new VQA benchmark dataset where questions are divided into 12 different categories based on the task they solve; 2) We propose two new evaluation metrics that compensate for forms of dataset bias; 3) We balance the number of yes/no object presence detection questions to assess whether a balanced distribution can help algorithms learn better; and 4) We introduce absurd questions that force an algorithm to determine if a question is valid for a given image. We then use the new dataset to re-train and evaluate both baseline and state-of-the-art VQA

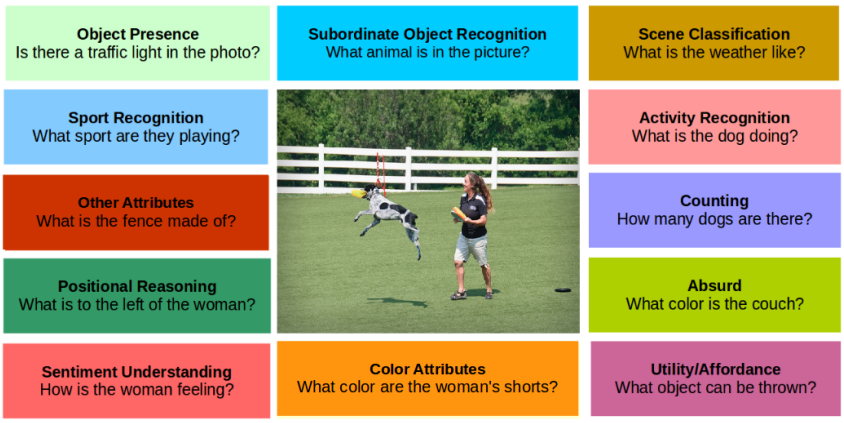



Figure 4.1: A good VQA benchmark tests a wide range of computer vision tasks in an unbiased manner. In this chapter, we propose a new dataset with 12 distinct tasks and evaluation metrics that compensate for bias, so that the strengths and limitations of algorithms can be better measured.

algorithms. We found that our proposed approach enables more nuanced comparisons of VQA algorithms, and helps us understand the benefits of specific techniques better. In addition, it also allowed us to answer several key questions about VQA algorithms, such as, 'Is the generalization capacity of the algorithms hindered by the bias in the dataset?', 'Does the use of spatial attention help answer specific question-types?', 'How successful are the VQA algorithms in answering less-common questions?', and 'Can the VQA algorithms differentiate between real and absurd questions?'

## 4.2 Background

### 4.2.1 Prior Natural Image VQA Datasets

Six datasets for VQA with natural images have been released between 2014–2016: DAQUAR [25], COCO-QA [71], FM-IQA [72], The VQA Dataset [12], Visual7W [73], and Visual Genome [74]. FM-IQA needs human judges and has not been widely used, so we do not discuss it further. Table 4.1 shows statistics for the other datasets. Following others [30, 75, 76], we refer to the portion of The VQA Dataset containing natural images as COCO-VQA. Detailed dataset reviews can be found in Chapter 2 and [133].

All of the aforementioned VQA datasets are biased. DAQUAR and COCO-QA are small and have a limited variety of question-types. Visual Genome, Visual7W, and COCO-

VQA are larger, but they suffer from several biases. Bias takes the form of both the kinds of questions asked and the answers that people give for them. For COCO-VQA, a system trained using only question features achieves 50% accuracy [30]. This suggests that some questions have predictable answers. While some of this is expected due to natural image statistics (E.g., a couch is more likely to co-occur with TV compared to a giraffe), if a system is unable to generalize to a rare set of conditions, it is not a robust or a trustworthy system. Without a more nuanced analysis, it is challenging to determine what kinds of questions are more dependent on the image. For datasets made using Mechanical Turk, annotators often ask object recognition questions, e.g., 'What is in the image?' or 'Is there an elephant in the image?'. Note that in the latter example, annotators rarely ask that kind of question unless the object is in the image. On COCO-VQA, 79% of questions beginning with 'Is there a' will have 'yes' as their ground truth answer.

In 2017, the VQA 2.0 [80] dataset was introduced. In VQA 2.0, the same question is asked for two different images and annotators are instructed to give opposite answers, which helped reduce language bias. However, in addition to language bias, these datasets are also biased in their distribution of different types of questions and the distribution of answers within each question-type. Existing VQA datasets use performance metrics that treat each test instance with equal value (e.g., simple accuracy). While some do compute additional statistics for basic question-types, overall performance is not computed from these sub-scores [12, 71]. This exacerbates the issues with the bias because the question-types that are more likely to be biased are also more common. Questions beginning with 'Why' and 'Where' are rarely asked by annotators compared to those beginning with 'Is' and 'Are'. For example, on COCO-VQA, improving accuracy on 'Is/Are' questions by 15% will increase overall accuracy by over 5%, but answering *all* 'Why/Where' questions correctly will increase accuracy by only 4.1% [13]. Due to the inability of the existing evaluation metrics to properly address these biases, algorithms trained on these datasets learn to exploit these biases, resulting in systems that work poorly when deployed in the real-world.

For related reasons, major benchmarks released in the last decade do not use simple accuracy for evaluating image recognition and related computer vision tasks, but instead use metrics such as mean-per-class accuracy that compensates for unbalanced categories. For example, on Caltech-101 [134], even with balanced training data, simple accuracy fails to address the fact that some categories were much easier to classify than others (e.g., faces and planes were easy and also had the largest number of test images). Mean per-class accuracy compensates for this by requiring a system to do well on each category, even when the amount of test instances in categories vary considerably.

Existing benchmarks do not require reporting accuracies across different question-types. Even when they are reported, the question-types can be too coarse to be useful, e.g., 'yes/no', 'number' and 'other' in COCO-VQA. To improve the analysis of the VQA algorithms, we categorize the questions into meaningful types, calculate the sub-scores,

and incorporate them in our evaluation metrics.

### 4.2.2 Synthetic Datasets that Fight Bias

Previous work has studied bias in VQA and proposed countermeasures. In [14], the Yin and Yang dataset was created to study the effect of having an equal number of binary (yes/no) questions about cartoon images. They found that answering questions from a balanced dataset was harder. This work is significant, but it was limited to yes/no questions and their approach using cartoon imagery cannot be directly extended to real-world images.

One of the goals of this work is to determine what kinds of questions an algorithm can answer easily. In [82], the SHAPES dataset was proposed, which has similar objectives. SHAPES is a small dataset, consisting of 64 images that are composed by arranging colored geometric shapes in different spatial orientations. Each image has the same 244 yes/no questions, resulting in 15,616 questions. Although SHAPES serves as an important adjunct evaluation, it alone cannot suffice for testing a VQA algorithm. The major limitation of SHAPES is that all of its images are of 2D shapes, which are not representative of real-world imagery. Along similar lines, Compositional Language and Elementary Visual Reasoning (CLEVR) [83] also proposes use of 3D rendered geometric objects to study reasoning capacities of a model. CLEVR is larger than SHAPES and makes use of 3D rendered geometric objects. In addition to shape and color, it adds material property to the objects. CLEVR has five types of questions: attribute query, attribute comparison, integer comparison, counting, and existence.

Both SHAPES and CLEVR were specifically tailored for compositional language approaches [82] and downplay the importance of visual reasoning. For instance, the CLEVR question, 'What size is the cylinder that is left of the brown metal thing that is left of the big sphere?' requires demanding language reasoning capabilities, but only limited visual understanding is needed to parse simple geometric objects. Unlike these three synthetic datasets, our dataset contains natural images and questions. To improve algorithm analysis and comparison, our dataset has more (12) explicitly defined question-types and new evaluation metrics.

## 4.3 TDIUC for Nuanced VQA Analysis

In past two years, multiple publicly released datasets have spurred the VQA research. However, due to the biases and issues with evaluation metrics, interpreting and comparing the performance of VQA systems can be opaque. We propose a new benchmark dataset that explicitly assigns questions into 12 distinct categories. This enables measuring performance within each category and understand which kind of questions are easy or hard

Table 4.1: Comparison of previous natural image VQA datasets with TDIUC. For COCO-VQA, the explicitly defined number of question-types is used, but a much finer granularity would be possible if they were individually classified. MC/OE refers to whether open-ended or multiple-choice evaluation is used.

| | **Images** | **Questions** | **Annotation Source** | **Question Types** | **Unique Answers** | **MC/OE** |
|---|---|---|---|---|---|---|
| **DAQUAR** | 1,449 | 16,590 | Both | 3 | 968 | OE |
| **COCO-QA** | 123,287 | 117,684 | Auto | 4 | 430 | OE |
| **COCO-VQA** | 204,721 | 614,163 | Manual | 3 | 145,172 | Both |
| **Visual7W** | 47,300 | 327,939 | Manual | 7 | 25,553 | MC |
| **Visual Genome** | 108,000 | 1,773,358 | Manual | 6 | 207,675 | OE |
| **TDIUC (Ours)** | 167,437 | 1,654,167 | Both | 12 | 1,618 | OE |

for today's best systems. Additionally, we use evaluation metrics that further compensate for the biases. We call the dataset the Task Driven Image Understanding Challenge (TDIUC). The overall statistics and example images of this dataset are shown in Table 4.1 and Fig. 4.2 respectively.

TDIUC has 12 question-types that were chosen to represent both classical computer vision tasks and novel high-level vision tasks which require varying degrees of image understanding and reasoning. The question-types are:

1. Object Presence (e.g., 'Is there a cat in the image?')
2. Subordinate Object Recognition (e.g., 'What kind of furniture is in the picture?')
3. Counting (e.g., 'How many horses are there?')
4. Color Attributes (e.g., 'What color is the man's tie?')
5. Other Attributes (e.g., 'What shape is the clock?')
6. Activity Recognition (e.g., 'What is the girl doing?')
7. Sport Recognition (e.g.,'What are they playing?')
8. Positional Reasoning (e.g., 'What is to the left of the man on the sofa?')
9. Scene Classification (e.g., 'What room is this?')
10. Sentiment Understanding (e.g.,'How is she feeling?')
11. Object Utilities and Affordances (e.g.,'What object can be used to break glass?')
12. Absurd (i.e., Nonsensical queries about the image)

The number of each question-type in TDIUC is given in Table 4.2. The questions come from three sources. First, we imported a subset of questions from COCO-VQA and Visual Genome. Second, we created algorithms that generated questions from COCO's semantic segmentation annotations [34], and Visual Genome's objects and attributes annotations [74]. Third, we used human annotators for certain question-types. In the following sections, we briefly describe each of these methods.


**Q:** What color is the suitcase? **A:** Absurd **Q:** What color is the man's hat? **A:** White **Q:** What sport is this? **A:** Tennis

**Q:** What is to the left of the blue bus? **A:** Car **Q:** Is there a train in the photo? **A:** No **Q:** How many bicycles are there? **A:** One

Figure 4.2: Images from TDIUC and their corresponding question-answer pairs.

### 4.3.1 Importing Questions from Existing Datasets

We imported questions from COCO-VQA and Visual Genome belonging to all question-types except 'object utilities and affordances'. We did this by using a large number of templates and regular expressions. For Visual Genome, we imported questions that had one word answers. For COCO-VQA, we imported questions with one or two word answers and in which five or more annotators agreed.

For color questions, a question would be imported if it contained the word 'color' in it and the answer was a commonly used color. Questions were classified as activity or sports recognition questions if the answer was one of nine common sports or one of fifteen common activities and the question contained common verbs describing actions or sports, e.g., playing, throwing, etc. For counting, the question had to begin with 'How many' and the answer had to be a small countable integer (1-16). The other categories were determined using regular expressions. For example, a question of the form 'Are ___feeling ___?' was classified as sentiment understanding and 'What is to the right of/left of/ behind the ___?' was classified as positional reasoning. Similarly, 'What `<OBJECT CATEGORY>` is in the image?' and similar templates were used to populate subordinate object recognition questions. This method was used for questions about the season and weather as well, e.g., 'What season is this?', 'Is this rainy/sunny/cloudy?', or 'What is the weather like?' were imported to scene classification.

### 4.3.2 Generating Questions using Image Annotations

Images in the COCO dataset and Visual Genome both have individual regions with semantic knowledge attached to them. We exploit this information to generate new questions using question templates. To introduce variety, we define multiple templates for each question-type and use the annotations to populate them. For example, for counting we use 8 templates, e.g., 'How many <objects> are there?', 'How many <objects> are in the photo?', etc. Since the COCO and Visual Genome use different annotation formats, we discuss them separately.

#### Questions Using COCO annotations

Sport recognition, counting, subordinate object recognition, object presence, scene understanding, positional reasoning, and absurd questions were created from COCO, similar to the scheme used in [17]. For **counting**, we count the number of object instances in an image annotation. To minimize ambiguity, this was only done if objects covered an area of at least 2,000 pixels.

For **subordinate object recognition**, we create questions that require identifying an object's subordinate-level object classification based on its larger semantic category. To do this, we use COCO supercategories, which are semantic concepts encompassing several objects under a common theme, e.g., the supercategory 'furniture' contains chair, couch, etc. If the image contains only one type of furniture, then a question similar to 'What kind of furniture is in the picture?' is generated because the answer is not ambiguous. Using similar heuristics, we create questions about identifying food, electronic appliances, kitchen appliances, animals, and vehicles.

For **object presence** questions, we find images with objects that have an area larger than 2,000 pixels and produce a question similar to 'Is there a <object> in the picture?' These questions will have 'yes' as an answer. To create negative questions, we ask questions about COCO objects that are not present in an image. To make this harder, we prioritize the creation of questions referring to absent objects that belong to the same supercategory of objects that are present in the image. A street scene is more likely to contain trucks and cars than it is to contain couches and televisions. Therefore, it is more difficult to answer 'Is there a truck?' in a street scene than it is to answer 'Is there a couch?'

For **sport recognition** questions, we detect the presence of specific sports equipment in the annotations and ask questions about the type of sport being played. Images must only contain sports equipment for one particular sport. A similar approach was used to create scene understanding questions. For example, if a toilet and a sink are present in annotations, the room is a bathroom and an appropriate scene recognition question can be created. Additionally, we use the supercategories 'indoor' and 'outdoor' to ask questions

about where a photo was taken.

For creating **positional reasoning** questions, we use the relative locations of bounding boxes to create questions similar to 'What is to the left/right of <object>?' This can be ambiguous due to overlapping objects, so we employ the following heuristics to eliminate ambiguity: 1) The vertical separation between the two bounding boxes should be within a small threshold; 2) The objects should not overlap by more than the half the length of its counterpart; and 3) The objects should not be horizontally separated by more than a distance threshold, determined by subjectively judging optimal separation to reduce ambiguity. We tried to generate above/below questions, but the results were unreliable.

**Absurd questions** test the ability of an algorithm to judge when a question is not answerable based on the image's content. To make these, we make a list of the objects that are absent from a given image, and then we find questions from rest of TDIUC that ask about these absent objects, with the exception of yes/no and counting questions. This includes questions imported from COCO-VQA, auto-generated questions, and manually created questions. We make a list of all possible questions that would be 'absurd' for each image and we uniformly sample three questions per image. In effect, we will have same question repeated multiple times throughout the dataset, where it can either be a genuine question or a nonsensical question. The algorithm must answer 'Does Not Apply' if the question is absurd.

#### Questions Using Visual Genome annotations

Visual Genome's annotations contain region descriptions, relationship graphs, and object boundaries. However, the annotations can be both non-exhaustive and duplicated, which makes using them to automatically make QA pairs difficult. We only use Visual Genome to make color and positional reasoning questions. The methods we used are similar to those used with COCO, but additional precautions were needed due to quirks in their annotations. Additional details are provided in the Appendix.

### 4.3.3 Manual Annotation

Creating sentiment understanding and object utility/affordance questions cannot be readily done using templates, so we used manual annotation to create these. Twelve volunteer annotators were trained to generate these questions, and they used a web-based annotation tool that we developed. They were shown random images from COCO and Visual Genome and could also upload images.

Table 4.2: The number of questions per type in TDIUC.

| | Questions | Unique Answers |
|---|---|---|
| Scene Recognition | 66,706 | 83 |
| Sport Recognition | 31,644 | 12 |
| Color Attributes | 195,564 | 16 |
| Other Attributes | 28,676 | 625 |
| Activity Recognition | 8,530 | 13 |
| Positional Reasoning | 38,326 | 1,300 |
| Sub. Object Recognition | 93,555 | 385 |
| Absurd | 366,654 | 1 |
| Utility/Affordance | 521 | 187 |
| Object Presence | 657,134 | 2 |
| Counting | 164,762 | 16 |
| Sentiment Understanding | 2,095 | 54 |
| Grand Total | 1,654,167 | 1,618 |

### 4.3.4 Post Processing

Post processing was performed on questions from all sources. All numbers were converted to text, e.g., 2 became two. All answers were converted to lowercase, and trailing punctuation was stripped. Duplicate questions for the same image were removed. All questions had to have answers that appeared at least twice. The dataset was split into train and test splits with 70% for train and 30% for test.

## 4.4 Proposed Evaluation Metric

One of the main goals of VQA research is to build computer vision systems capable of many tasks, instead of only having expertise at one specific task (e.g., object recognition). For this reason, some have argued that VQA is a kind of Visual Turing Test [25]. However, if simple accuracy is used for evaluating performance, then it is hard to know if a system succeeds at this goal because some question-types have far more questions than others. In VQA, skewed distributions of question-types are to be expected. If each test question is treated equally, then it is difficult to assess performance on rarer question-types and to compensate for bias. We propose multiple measures to compensate for bias and skewed distributions.

To compensate for the skewed question-type distribution, we compute accuracy for each of the 12 question-types separately. However, it is also important to have a final unified accuracy metric. Our overall metrics are the arithmetic and harmonic means across all per question-type accuracies, referred to as arithmetic mean-per-type (Arithmetic MPT) accuracy and harmonic mean-per-type accuracy (Harmonic MPT). Unlike the Arithmetic

MPT, Harmonic MPT measures the ability of a system to have high scores across all question-types and is skewed towards lowest performing categories.

We also use normalized metrics that compensate for bias in the form of imbalance in the distribution of answers within each question-type, e.g., the most repeated answer 'two' covers over 35% of all the counting-type questions. To do this, we compute the accuracy for each unique answer separately within a question-type and then average them together for the question-type. To compute overall performance, we compute the arithmetic normalized mean per-type (N-MPT) and harmonic N-MPT scores. A large discrepancy between unnormalized and normalized scores suggests an algorithm is not generalizing to rarer answers.

## 4.5 Algorithms for VQA tested on TDIUC

Firstly, we consider a simple baseline model: a simple multi-layer perceptron (MLP) classifier that take as input the question and image embeddings concatenated to each other [12, 30, 76], where the image features come from the last hidden layer of a CNN. These simple approaches often work well and can be competitive with complex attentive models [30, 76].

Next, we consider an algorithm that uses spatial attention. Specifically, we consider the MCB algorithm [88] which won the CVPR-2016 VQA Workshop Challenge. Spatial attention allows algorithms to weigh the relevance of different image regions for a given question. For example, to answer 'What color is the bear?' they aim emphasize the visual features around the bear and suppress other features. Since TDIUC consists of nuanced categories, we can study whether attention is more effective for some categories than others.

Next, we consider neural module network (NMN) is an especially interesting compositional approach to VQA [82, 92]. The main idea is to compose a series of discrete modules (sub-networks) that can be executed collectively to answer a given question. We include this model to better study whether a compositional approach fares better under the different question-types in TDIUC.

Finally, we consider multi-step recurrent answering units (RAU) model [97] as another high-performing VQA algorithm that performs on par with the MCB model on the existing datasets. We include RAU to study whether TDIUC can reveal new insights about algorithms that perform similarly on existing datasets.

All of these algorithms were described in more detail in Chapter 2.

Table 4.3: Results for all VQA models. The unnormalized accuracy for each question-type is shown. Normalized scores for individual question-types are presented in the appendix table A.2. * denotes training without absurd questions.

| | **YES** | **REP** | **IMG** | **QUES** | **Q+I** | ***Q+I** | **MLP** | **MCB** | ***MCB** | **MCB-A** | **NMN** | **RAU** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scene Recognition | 26.90 | 26.90 | 14.25 | 53.18 | 72.19 | 72.75 | 91.45 | 92.04 | 91.87 | 93.06 | 91.88 | **93.96** |
| Sport Recognition | 0.00 | 22.05 | 18.61 | 18.87 | 85.16 | 89.40 | 90.24 | 92.47 | 92.47 | 92.77 | 89.99 | **93.47** |
| Color Attributes | 0.00 | 22.74 | 0.92 | 37.60 | 43.69 | 50.52 | 53.64 | 56.93 | 57.07 | **68.54** | 54.91 | 66.86 |
| Other Attributes | 0.00 | 24.23 | 2.07 | 36.13 | 42.89 | 51.47 | 41.79 | 53.24 | 54.62 | **56.72** | 47.66 | 56.49 |
| Activity Recognition | 0.00 | 21.63 | 3.06 | 10.81 | 24.16 | 48.55 | 39.22 | 51.42 | **53.58** | 52.35 | 44.26 | 51.60 |
| Positional Reasoning | 0.00 | 6.05 | 2.23 | 14.23 | 25.15 | 27.73 | 21.87 | 33.34 | 33.02 | **35.40** | 27.92 | 35.26 |
| Sub. Object Recognition | 0.00 | 7.16 | 10.55 | 21.40 | 80.92 | 81.66 | 80.55 | 84.63 | 84.58 | 85.54 | 82.02 | **86.11** |
| Absurd | 0.00 | **100.00** | 19.97 | 96.71 | 96.98 | N/A | 95.96 | 83.44 | N/A | 84.82 | 87.51 | 96.08 |
| Utility and Affordances | 11.70 | 11.70 | 5.26 | 16.37 | 24.56 | 30.99 | 13.45 | 33.92 | 29.24 | **35.09** | 25.15 | 31.58 |
| Object Presence | 50.00 | 50.00 | 20.73 | 69.06 | 69.43 | 69.50 | 92.33 | 91.84 | 91.55 | 93.64 | 92.50 | **94.38** |
| Counting | 0.00 | 36.19 | 0.30 | 44.51 | 44.82 | 44.84 | 51.12 | 50.29 | 50.07 | **51.01** | 49.21 | 48.43 |
| Sentiment Understanding | 44.64 | 44.64 | 15.93 | 52.84 | 53.00 | 59.94 | 58.33 | 65.46 | 66.25 | **66.25** | 58.04 | 60.09 |
| Overall (Arithmetic MPT) | 11.10 | 31.11 | 9.49 | 39.31 | 55.25 | 57.03 | 60.87 | 65.75 | 66.07 | **67.90** | 62.59 | 67.81 |
| Overall (Harmonic MPT) | 0.00 | 17.53 | 1.92 | 25.93 | 44.13 | 50.30 | 42.80 | 58.03 | 55.43 | **60.47** | 51.87 | 59.00 |
| Overall (Arithmetic N-MPT) | 4.87 | 15.63 | 5.82 | 21.46 | 29.47 | 28.10 | 31.36 | 39.81 | 35.49 | **42.24** | 34.00 | 41.04 |
| Overall (Harmonic N-MPT) | 0.00 | 0.83 | 1.91 | 8.42 | 14.99 | 18.30 | 9.46 | 24.77 | 23.20 | **27.28** | 16.67 | 23.99 |
| Simple Accuracy | 21.14 | 51.15 | 14.54 | 62.74 | 69.53 | 63.30 | 81.07 | 79.20 | 78.06 | 81.86 | 79.56 | **84.26** |

## 4.6 Experiments

We trained multiple baseline models as well as state-of-the-art VQA methods on TDIUC. The methods we use are:

- **YES**: Predicts 'yes' for all questions.
- **REP**: Predicts the most repeated answer in a question-type category using an oracle.
- **QUES**: A linear softmax classifier given only question features (image blind).
- **IMG**: A linear softmax classifier given only image features (question blind).
- **Q+I**: A linear classifier given the question and image..
- **MLP**: A 4-layer MLP fed question and image features.
- **MCB**: MCB [88] without spatial attention.
- **MCB-A**: MCB [88] with spatial attention.
- **NMN**: NMN from [82] with minor modifications.
- **RAU**: RAU [97] with minor modifications.

For image features, ResNet-152 [4] with $448 \times 448$ images was used for all models.

QUES and IMG provide information about biases in the dataset. QUES, Q+I, and MLP all use 4800-dimensional skip-thought vectors [114] to embed the question, as was done in [30]. For image features, these all use the 'pool5' layer of ResNet-152 normalized to unit length. MLP is a 4-layer net with a softmax output layer. The 3 ReLU hidden layers have 6000, 4000, and 2000 units, respectively. During training, dropout (0.3) was used for the hidden layers.

For MCB, MCB-A, NMN and RAU, we used publicly available code to train them on TDIUC. The experimental setup and hyperparamters were kept unchanged from the default choices in the code, except for upgrading NMN and RAU's visual representation to both use ResNet-152.

Results on TDIUC for these models are given in Table 4.3. Accuracy scores are given for each of the 12 question-types in Table 4.3, and scores that are normalized by using mean-per-unique-answer are given in appendix Table A.2.

## 4.7 Detailed Analysis of VQA Models

### 4.7.1 Easy Question-Types for Today's Methods

By inspecting Table 4.3, we can see that some question-types are comparatively easy ($> 90\%$) under MPT: scene recognition, sport recognition, and object presence. High accuracy is also achieved on absurd, which we discuss in greater detail in Sec. 4.7.4. Sub-ordinate object recognition is moderately high ($> 80\%$), despite having a large number of unique answers. Accuracy on counting is low across all methods, despite a large number of training data. For the remaining question-types, more analysis is needed to pinpoint

whether the weaker performance is due to lower amounts of training data, bias, or limitations of the models. We next investigate how much of the good performance is due to bias in the answer distribution, which N-MPT compensates for.

### 4.7.2 Effects of the Proposed Accuracy Metrics

One of our major aims was to compensate for the fact that algorithms can achieve high scores by simply learning to answer more populated and easier question-types. For existing datasets, earlier work has shown that simple baseline methods routinely exceed more complex methods using simple accuracy [30, 76, 112]. On TDIUC, MLP surpasses MCB and NMN in terms of simple accuracy, but a closer inspection reveals that MLP's score is highly determined by performance on categories with a large number of examples, such as 'absurd' and 'object presence.' Using MPT, we find that both NMN and MCB outperform MLP. Inspecting normalized scores for each question-type (Appendix Table A.2) shows an even more pronounced differences, which is also reflected in arithmetic N-MPT score presented in Table 4.3. This indicates that MLP is prone to overfitting. Similar observations can be made for MCB-A compared to RAU, where RAU outperforms MCB-A using simple accuracy, but scores lower on *all* the metrics designed to compensate for the skewed answer distribution and bias.

Comparing the unnormalized and normalized metrics can help us determine the generalization capacity of the VQA algorithms for a given question-type. A large difference in these scores suggests that an algorithm is relying on the skewed answer distribution to obtain high scores. We found that for MCB-A, the accuracy on subordinate object recognition drops from 85.54% with unnormalized to 23.22% with normalized, and for scene recognition it drops from 93.06% (unnormalized) to 38.53% (normalized). Both these categories have a heavily skewed answer distribution; the top-25 answers in subordinate object recognition and the top-5 answers in scene recognition cover over 80% of all questions in their respective question-types. This shows that question-types that appear to be easy may simply be due to the algorithms learning the answer statistics. A truly easy question-type will have similar performance for both unnormalized and normalized metrics. For example, sport recognition shows only 17.39% drop compared to a 30.21% drop for counting, despite counting having same number of unique answers and far more training data. By comparing relative drop in performance between normalized and unnormalized metric, we can also *compare* the generalization capability of the algorithms, e.g., for subordinate object recognition, RAU has higher unnormalized score (86.11%) compared to MCB-A (85.54%). However, for normalized scores, MCB-A has significantly higher performance (23.22%) than RAU (21.67%). This shows RAU may be more dependent on the answer distribution. Similar observations can be made for MLP compared to MCB.

### 4.7.3 Can Algorithms Predict Rare Answers?

In the previous section, we saw that the VQA models struggle to correctly predict rarer answers. Are the less repeated questions *actually* harder to answer, or are the algorithms simply biased toward more frequent answers? To study this, we created a subset of TDIUC that only consisted of questions that have answers repeated less than 1000 times. We call this dataset TDIUC-Tail, which has 46,590 train and 22,065 test questions. Then, we trained MCB on: 1) the full TDIUC dataset; and 2) TDIUC-Tail. Both versions were evaluated on the validation split of TDIUC-Tail.

We found that MCB trained only on TDIUC-Tail outperformed MCB trained on all of TDIUC across all question-types (details are in appendix Table A.3 and A.4). This shows that MCB is capable of learning to correctly predict rarer answers, but it is simply biased towards predicting more common answers to maximize overall accuracy. Using normalized accuracy disincentivizes the VQA algorithms' reliance on the answer statistics, and for deploying a VQA system it may be useful to optimize directly for N-MPT.

It should be noted that the version trained only on TDIUC-Tail does not perform well on the full TDIUC dataset. Therefore this experiment should not be construed as a valid solution to predicting rare answers. Rather, this experiment shows that the questions in the tail are not *inherently* more difficult than the majority questions, but rather are made difficult to learn by the how VQA algorithms are currently trained.

### 4.7.4 Effects of Including Absurd Questions

Absurd questions force a VQA system to look at the image to answer the question. In TDIUC, these questions are sampled from the rest of the dataset, and they have a high prior probability of being answered 'Does not apply.' This is corroborated by the QUES model, which achieves a high accuracy on absurd; however, for the same questions when they are genuine for an image, it only achieves 6.77% accuracy on these questions. Good absurd performance is achieved by sacrificing performance on other categories. A robust VQA system should be able to detect absurd questions without then failing on others. By examining the accuracy on real questions that are identical to absurd questions, we can quantify an algorithm's ability to differentiate the absurd questions from the real ones. We found that simpler models had much lower accuracy on these questions, (QUES: 6.77%, Q+I: 34%), compared to more complex models (MCB: 62.44%, MCB-A: 68.83%).

To further study this, we we trained two VQA systems, Q+I and MCB, both with and without absurd. The results are presented in Table 4.3. For Q+I trained without absurd questions, accuracies for other categories increase considerably compared to Q+I trained with full TDIUC, especially for question-types that are used to sample absurd questions, e.g., activity recognition (24% when trained with absurd and 48% without). Arithmetic MPT accuracy for the Q+I model that is trained without absurd (57.03%) is also sub-

stantially greater than MPT for the model trained with absurd (51.45% for all categories except absurd). This suggests that Q+I is not properly discriminating between absurd and real questions and is biased towards mis-identifying genuine questions as being absurd. In contrast, MCB, a more capable model, produces worse results for absurd, but the version trained without absurd shows much smaller differences than Q+I, which shows that MCB is more capable of identifying absurd questions.

### 4.7.5 Effects of Balancing Object Presence

In Sec. 4.7.3, we saw that a skewed answer distribution can impact generalization. This effect is strong even for simple questions and affects even the most sophisticated algorithms. Consider MCB-A when it is trained on both COCO-VQA and Visual Genome, i.e., the winner of the CVPR-2016 VQA Workshop Challenge. When it is evaluated on object presence questions from TDIUC, which contains 50% 'yes' and 50% 'no' questions, it correctly predicts 'yes' answers with 86.3% accuracy, but only 11.2% for questions with 'no' as an answer. However, after training it on TDIUC, MCB-A is able to achieve 95.02% for 'yes' and 92.26% for 'no.' MCB-A performed poorly by learning the biases in the COCO-VQA dataset, but it is capable of performing well when the dataset is unbiased. Similar observations about balancing yes/no questions were made in [14]. Datasets could balance simple categories like object presence, but extending the same idea to all other categories is a challenging task and undermines the natural statistics of the real-world. Adopting mean-per-class and normalized accuracy metrics can help compensate for this problem.

### 4.7.6 Advantages of Attentive Models

By breaking questions into types, we can assess which types benefit the most from attention. We do this by comparing the MCB model with and without attention, i.e., MCB and MCB-A. As seen in Table 4.3, attention helped improve results on several question categories. The most pronounced increases are for color recognition, attribute recognition, absurd, and counting. All of these question-types require the algorithm to detect specified object(s) (or lack thereof) to be answered correctly. MCB-A computes attention using local features from different spatial locations, instead of global image features. This aids in localizing individual objects. The attention mechanism learns the relative importance of these features. RAU also utilizes spatial attention and shows similar increments.

### 4.7.7 Compositional and Modular Approaches

NMN, and, to a lesser extent, RAU propose compositional approaches for VQA. For COCO-VQA, NMN has performed worse than some MLP models [30] using simple ac-

curacy. We hoped that it would achieve better performance than other models for questions that require logically analyzing an image in a step-by-step manner, e.g., positional reasoning. However, while NMN did perform better than MLP using MPT and N-MPT metric, we did not see any substantial benefits in specific question-types. This may be because NMN is limited by the quality of the 'S-expression' parser, which produces incorrect or misleading parses in many cases. For example, 'What color is the jacket of the man on the far left?' is parsed as `(color jacket);(color leave);(color (and jacket leave))`. This expression not only fails to parse 'the man', which is a crucial element needed to correctly answer the question, but also wrongly interprets 'left' as past tense of leave.

RAU performs inference over multiple hops, and because each hop contains a complete VQA system, it can learn to solve different tasks in each step. Since it is trained end-to-end, it does not need to rely on rigid question parses. It showed very good performance in detecting absurd questions and also performed well on other categories.

## 4.8 Conclusion

We introduced TDIUC, a VQA dataset that consists of 12 explicitly defined question-types, including absurd questions, and we used it to perform a rigorous analysis of recent VQA algorithms. We proposed new evaluation metrics to compensate for biases in VQA datasets. Results show that the absurd questions and the new evaluation metrics enable a deeper understanding of VQA algorithm behavior.

# Chapter 5

# DVQA: Understanding Data Visualizations via Question Answering

## 5.1 Introduction

Data visualizations, *e.g.* , bar charts, pie charts, and plots, contain large amounts of information in a concise format. These visualizations are specifically designed to communicate data to people, and are not designed to be machine interpretable. Nevertheless, making algorithms capable to make inferences from data visualizations has enormous practical applications. Here, we study systems capable of answering open-ended questions about bar charts, which we refer to as data visualization question answering (DVQA). DVQA would enable vast repositories of charts within scientific documents, web-pages, and business reports to be queried automatically. Automatically parsing bar-charts can also enable visually impaired users to navigate documents with charts in them, since most documents do not include sufficient alt tags and metadata to allow the contents of bar-charts to be parsed [135]. Example DVQA images and questions grouped by the different tasks are shown in Fig. 5.1.

Besides practical benefits, DVQA can also serve as a challenging proxy task for generalized pattern matching, attention, and multi-step reasoning systems. Answering a question about a chart requires multi-step attention, memory, measurement, and reasoning that poses significant challenges to the existing systems. For example, to answer the question *'What is the accuracy of algorithm* vice *on the dataset* fear*?'* in Fig. 5.1 can require finding the appropriate color and hatching that represents the dataset *fear*, finding the group of bars that represent the algorithm *vice*, measuring the height of the bar based on the y-axis, and if necessary interpolating between two neighboring values.

DVQA is related to visual question answering (VQA) [12, 25], which deals with answering open-ended questions about images. VQA is usually treated as a classification problem, in which answers are categories that are inferred using features from image-

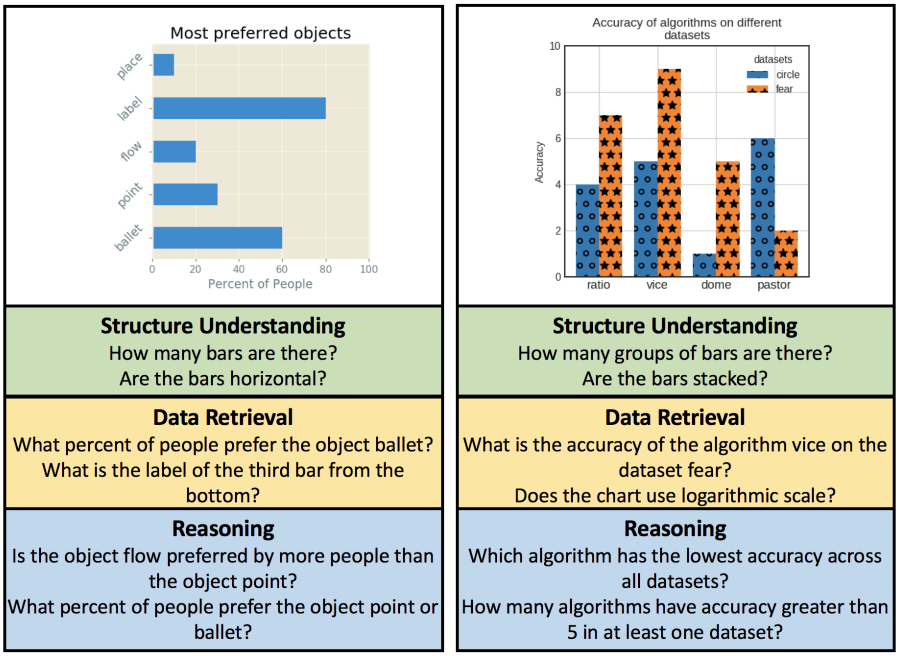



Figure 5.1: DVQA involves answering questions about diagrams. We present a dataset for DVQA with bar charts that exhibit enormous variety in appearance and style. We show that VQA systems cannot answer many DVQA questions and we describe more effective algorithms.

question pairs. DVQA poses three major challenges that are overlooked by existing VQA datasets with natural images. First, VQA systems typically assume two fixed vocabulary dictionaries: one for encoding words in questions and one for producing answers. In DVQA, assuming a fixed vocabulary makes it impossible to properly process many questions or to generate answers unique to a bar chart, which are often labeled with proper nouns, abbreviations, or concatenations (*e.g.* , 'Jan-Apr'). Our models demonstrate two ways for handling out-of-vocabulary (OOV) words. Second, the language utilized in VQA systems represent fixed semantic concepts that are immutable over images, *e.g.* , phrases such as 'A large shiny red cube' used in CLEVR [83] represent a fixed concept; once the word 'red' is associated with the underlying semantic concept, it is immutable. By contrast, the words utilized in labels and legends in DVQA can often be arbitrary and could refer to bars of different position, size, texture, and color. Third, VQA's natural images exhibit regularities that are not present in DVQA, *e.g.* to infer the answer to 'What is the weather like?' for the image in Fig. 5.2, an agent could use color tones and overall brightness to infer 'sunny.' Changing the color of the fire hydrant only changes the local information that impacts questions about the fire hydrant's properties. However, in bar charts, a small change, *e.g.* , shuffling the colors of the legend in Fig. 5.2, completely

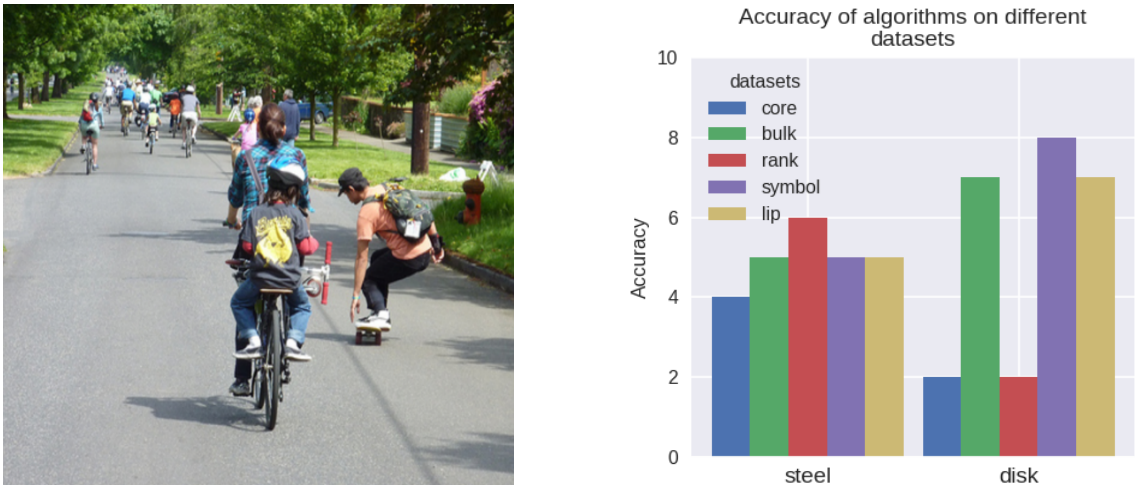



Figure 5.2: Natural images vs. bar charts. **Left**: Small changes in an image typically have little impact on a question in VQA. **Right:** Bar charts convey information using a sparse, but precise, set of visual elements. Even small changes can completely alter the information in the chart.

alters the chart's information. This makes DVQA an especially challenging problem.

**This chapter makes three major contributions:**

1. We describe the DVQA dataset, which contains over 3 million image-question pairs about bar charts. It tests three forms of diagram understanding: a) structure understanding; b) data retrieval; and c) reasoning. The DVQA dataset will be publicly released.
2. We show that both baseline and state-of-the-art VQA algorithms are incapable of answering many of the questions in DVQA. Moreover, existing classification based systems based on a static and predefined vocabulary are incapable of answering questions with unique answers that are not encountered during training.
3. We describe two DVQA systems capable of handling words that are unique to a particular image. One is an end-to-end neural network that can read answers from the bar chart. The second is a model that encodes a bar chart's text using a dynamic local dictionary.

## 5.2 Related Work

### 5.2.1 Automatically Parsing Bar Charts

Extracting data from bar charts using computer vision has been extensively studied [136–140]. Some focus on extracting the visual elements from the bar charts [139], while others

focus on extracting the data from each bar directly [138, 140]. Most of these approaches use fixed heuristics and make strong simplifying assumptions, *e.g.* , [140] made several simplifying assumptions about bar chart appearance (bars are solidly shaded without textures or gradients, no stacked bars, etc.). Moreover, they only tested their data extraction procedure on a total of 41 bar charts.

Our DVQA dataset has variations in bar chart appearance that go far beyond the capabilities of any of the aforementioned works. Moreover, DVQA requires more than just data extraction. Correctly answering DVQA questions requires basic language understanding, attention, concept of working short-term memory and reasoning.

### 5.2.2 VQA with Natural Images

DVQA, by design, is closely linked with VQA for natural images, which we have extensively discussed in previous chapters. While there are significant similarities between VQA and DVQA, one critical difference is that many DVQA questions require directly reading text from a chart to correctly answer them. This demands being able to handle words that are unique to a particular chart, which is a capability that is not needed by algorithms operating on existing VQA datasets with natural images.

### 5.2.3 Reasoning, Synthetic Scenes, and Diagrams

While VQA is primarily studied using natural images, several datasets have been proposed that use synthetic scenes or diagrams to test reasoning and understanding [83, 141, 142]. The CLEVR [83] dataset has complex reasoning questions about synthetically created scenes, and systems that perform well on popular VQA datasets perform poorly on CLEVR. The TQA [142] and AI2D [141] datasets both involve answering science questions about text and images. Both datasets are relatively small, *e.g.* , AI2D only contains 15,000 questions. These datasets require more than simple pattern matching and memorization. Similar to our work, their creators showed that state-of-the-art VQA systems for natural image datasets performed poorly on their datasets. However, there are key differences between these datasets and DVQA. First, none of these datasets contain questions specific to bar charts. Second, their datasets use multiple-choice schemes that reduce the problem to a ranking problem, rather than the challenges posed by having to generate open-ended answers. Studies have shown that multiple-choice schemes have biases that models will learn to exploit [112]. In contrast, we treat DVQA as an open-ended question answering task.

Concurrent to our work, FigureQA [2] also explores question answering for charts, however, with following major limitations compared to our DVQA dataset: 1) it contains only yes/no type questions; 2) it does not contain questions that require numeric values as

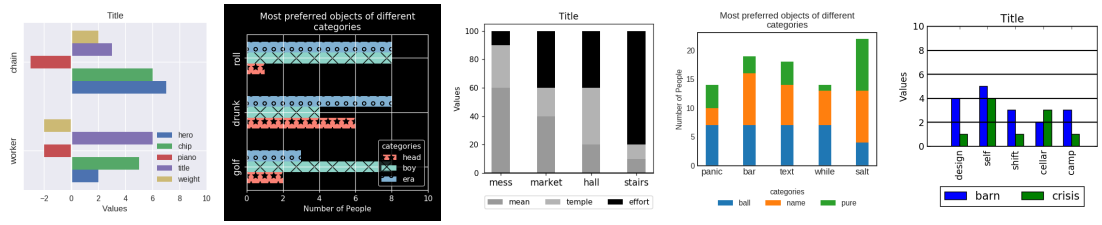



Figure 5.3: Example bar chart images from DVQA. DVQA contains significant variation in appearance and style.

answers; 3) it has fixed labels for bars across different figures (*e.g.* , a red bar is always labeled 'red'); and 4) it avoids the OOV problem.

## 5.3 DVQA: The Dataset

DVQA is a challenging synthetic dataset that tests multiple aspects of bar chart understanding that cause state-of-the-art methods for VQA to fail, which we demonstrate in experiments. Synthetically generating DVQA gave us precise control over the positions and appearances of the visual elements. It also gave us access to meta-data about these components, which would not be available with real data. This meta-data contains all information within the chart, including the precise position of each drawing element, the underlying data used to create the chart and location of all text-elements. This data can be used as an additional source of supervision or to ensure that an algorithm is "attending" to relevant regions. As shown in Fig. 5.3, the DVQA dataset contains a large variety of typically available styles of bar chart. The questions in the dataset require the ability to reason about the information within a bar chart (see Fig. 5.1). DVQA contains 3,487,194 total question answer pairs for 300,000 images divided into three major question types. Tables 5.1 and 5.2 show statistics about the DVQA dataset. Additional statistics are given in the supplemental materials.

### 5.3.1 Appearance, Data, and Question Types

DVQA consists of bar charts with question-answer pairs that are generated by selecting a visual style for a chart, choosing data for a chart, and then generating questions for that chart. Here, we briefly explain how this was done. Additional details are provided in the supplemental materials.

**Visual Styles:** We use python's popular drawing tool, Matplotlib to generate our charts since it offers unparalleled programmatic control over each of the element drawn. As shown in Fig. 5.3, DVQA's bar charts contain a wide variability in both appearance and

style that can capture the common styles found in scientific documents and the Internet. Some of these variations include the difference in the number of bars and groups; presence or absence of grid lines; difference in color, width, spacing, orientation, and texture of the bars; and difference in the orientation and the location of labels and legends.

To label individual bars and legend entries, we select the 1000 most frequent nouns in the Brown Corpus using NLTK's part-of-speech tagging for our training set and our 'easy' test set ***Test-Familiar***. To measure a system's ability to scale to unknown answers, we also created a more difficult test set ***Test-Novel***, in which we use 500 new words that are not seen during training.

**Underlying Data:** DVQA has three bar chart data types: linear, percentage, and exponential. For each of these data value types, the bars can take any of the 10 randomly chosen values in the range 1 – 10 for linear data, 10 – 100 for percentage, and 1 - $10^{10}$ for exponential data type. A small percentage of bars are allowed to have a value of zero which appears as a missing bar in the chart.

**Question Types:** DVQA contains three types of questions: 1) structure understanding, 2) data retrieval, and 3) reasoning. To generate these questions, we use fixed templates. Based on the context of the chart reflected through its title and labels, the questions will vary along the template. Below, we will show a random assortment of these questions with further details presented in the supplementary materials.

**Structure Understanding.** Structure understanding questions test a system's ability to understand the overall structure of a bar chart. These questions include:

1. How many bars are there?
2. How many groups/stacks of bars are there?
3. How many bars are there per group?
4. Does the chart contain any negative values?
5. Are the bars horizontal?
6. Does the chart contain stacked bars?
7. Is each bar a single solid color without patterns?

**Data Retrieval.** Data retrieval questions test a system's ability to retrieve information from a bar chart by parsing the chart into its individual components. These questions often require paying attention to specific region of the chart. These questions include:

1. Are the values in the chart presented in a logarithmic scale?
2. Are the values in the chart presented in a percentage scale?
3. What percentage of people prefer the object **O**?
4. What is the label of the third bar from the left?
5. What is the label of the first group of bars from the left?
6. What is the label of the second bar from the left in each group?
7. What element does the **C** color represent?
8. How many units of the item **I** were sold in the store **S**?

Table 5.1: Dataset statistics for different DVQA splits for different question types.

| | **Total Questions** | **Unique Answers** |
|---|---|---|
| Structure | 471,108 | 10 |
| Data | 1,113,704 | 1,538 |
| Reasoning | 1,613,974 | 1,576 |
| **Grand Total** | **3,487,194** | **1,576** |

**Reasoning.** Reasoning questions test a model's ability to collect information from multiple components of a bar chart and perform operations on them. These include:

1. Which algorithm has the highest accuracy?
2. How many items sold more than $N$ units?
3. What is the difference between the largest and the smallest value in the chart?
4. How many algorithms have accuracies higher than **N**?
5. What is the sum of the values of **L1** and **L2**?
6. Did the item **I1** sold less units than **I2**?
7. How many groups of bars contain at least one bar with value greater than N?
8. Which item sold the most units in any store?
9. Which item sold the least number of units summed across all the stores?
10. Is the accuracy of the algorithm **A1** in the dataset **D1** larger than the accuracy of the algorithm **A2** in the dataset **D2**?

### 5.3.2 Post-processing to Minimize Bias

Several studies in VQA have shown that bias in datasets can impair performance evaluation and give inflated scores to systems that simply exploit statistical patterns [16,17,112]. In DVQA, we have taken several measures to combat such biases. To ensure that there is no correlation between styles, colors, and labels, we randomize the generation of charts. Some questions can have strong priors, *e.g.* , the question 'Does the chart contain stacked bar?' has a high probability of the correct answer being 'no' because these stacked charts are uncommon. To compensate for this, we randomly remove these questions until yes/no answers are balanced for each question type where yes/no is an answer. A similar scheme was used to balance other structure understanding questions as well as the first two data retrieval questions.

## 5.4 DVQA Algorithms & Models

In this section, we describe two novel deep neural network algorithms along with five baselines. Our proposed algorithms are able to read text from bar charts, giving them the

Table 5.2: DVQA dataset statistics for different splits.

| | Images | Questions | Unique Answers |
|---|---|---|---|
| Train | 200,000 | 2,325,316 | 1,076 |
| Test-Familiar | 50,000 | 580,557 | 1,075 |
| Test-Novel | 50,000 | 581,321 | 577 |
| **Grand Total** | **300,000** | **3,487,194** | **1,576** |

ability to answer questions with chart-specific answers or requiring chart-specific information.

All of the models that process images use the ImageNet pre-trained ResNet-152 [4] CNN with $448 \times 448$ images resulting in a $14 \times 14 \times 2048$ feature tensor, unless otherwise noted. All models that process questions use a 1024 unit single layer LSTM to encode questions, where each word in the question is embedded into a dense 300 dimensional representation. Training details are given in Sec. 5.4.4.

### 5.4.1 Baseline Models

We evaluate five baseline models for DVQA:

1. **YES**: This model answers 'YES' for all questions, which is the most common answer in DVQA by a small margin over 'NO'.
2. **IMG**: A question-blind model. Images are encoded using Resnet using the output of its final convolutional layer after pooling, and then the answer is predicted from them by an MLP with one hidden-layer that has 1,024 units and a softmax output layer.
3. **QUES**: An image-blind model. It uses the LSTM encoder to embed the question, and then the answer is predicted by an MLP with one hidden-layer that has 1,024 units and a softmax output layer.
4. **IMG+QUES**: This is a combination of the QUES and IMG models. It concatenates the LSTM and CNN embeddings, and then feeds them to an MLP with one 1024-unit hidden layer and a softmax output layer.
5. **SAN-VQA**: The Stacked Attention Network (SAN) [91] for VQA. We use our own implementation of SAN as described by [124], where it was shown that upgrading the original SAN's image features and a couple small changes produces state-of-the-art results on VQA 1.0 and 2.0. SAN operates on the last CNN convolutional feature maps, where it processes this map attentively using the question embedding from our LSTM-based scheme.

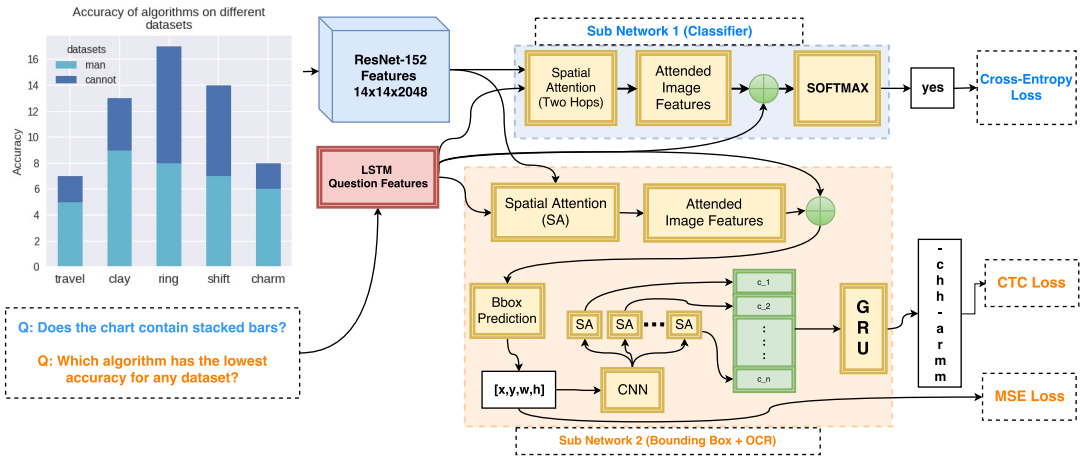



Figure 5.4: Overview of our Multi-Output Model (MOM) for DVQA. MOM uses two sub-networks: 1) classification sub-network that is responsible for *generic answers*, and 2) OCR sub-network that is responsible for *chart-specific answers*.

### 5.4.2 Multi-Output Model (MOM)

Our Multi-Output Model (MOM) for DVQA uses a dual-network architecture, where one of its sub-networks is able to generate chart-specific answers. MOM's classification sub-network is responsible for *generic* answers. MOM's optical character recognition (OCR) sub-network is responsible for *chart-specific* answers that must be read from the bar chart. The classification sub-network is identical to the SAN-VQA algorithm described earlier in Sec. 5.4.1. An overview is given in Fig. 5.4.

MOM's OCR sub-network tries to predict the bounding box containing the correct label and then applies a character-level decoder to that region. The bounding box predictor is trained as a regression task using a mean-squared-error (MSE) loss. An image patch is extracted from this region, which is resized to $128 \times 128$, and then a small 3-layer CNN is applied to it. Since the orientation of the text in the box will vary, we employ an $N$-step spatial attention mechanism to encode the relevant features for each of the $N$ possible characters in the image patch, where $N$ is the largest possible character-sequence ($N = 8$ in our experiments). These $N$ features are encoded using a bi-directional gated recurrent unit (GRU) to capture the character level correlations found in naturally occurring words. The GRU encoding is followed by a classification layer that predicts the character sequence, which is trained using connectionist temporal classification (CTC) loss [143].

MOM must determine whether to use the classification sub-network (*i.e.* SAN-VQA) or the OCR sub-network to answer a question. To determine this, we train a separate binary classifier that determines which of the outputs to trust. This classifier takes the LSTM question features as input to predict whether the answer is generic or chart-specific.

For our DVQA dataset this classifier is able to predict the correct branch with perfect accuracy on the test data.

### 5.4.3 SANDY: SAN with DYnamic Encoding Model

MOM handles chart-specific answers by having a sub-network capable of generating unique strings; however, it has no explicit ability to visually read bar chart text and its LSTM question encoding cannot handle chart-specific words. To explore overcoming these limitations, we modified SAN to create SANDY, SAN with DYnamic encoding model. SANDY uses a dynamic encoding model (DEM) that explicitly encodes chart-specific words in the question, and can directly generate chart-specific answers. The DEM is a dynamic local dictionary for chart-specific words. This dictionary is used for encoding words as well as answers.

To create a local word dictionary, DEM assumes it has access to an OCR system that gives it the positions and strings for all text-areas in a bar chart. Given this collection of boxes, DEM assigns each box a unique numeric index. It assigns an index of $0$ to the box in the lower-left corner of the image. Then, it assigns the box with the position closest to the first box with an index of $1$. The box closest to $1$ that is not yet assigned an index is then assigned the index of $2$, and so on until all boxes in the image are assigned an index. In our implementation, we assume that we have a perfect (oracle) OCR system for input, and we use the dataset's annotations for this purpose. No chart in the training data had more than 30 text labels, so we set the local dictionary to have at most $M = 30$ elements.

The local dictionary augments the $N$ element global dictionary. This enables DEM to create $(M + N)$-word dictionary that are used to encode each word in a question. The local dictionary is also used to augment the $L$ element global answer dictionary. This is done by adding $M$ extra classes to the classifier representing the dynamic words. If these classes are predicted, then the output string is assigned using the local dictionary's appropriate index.

We test two versions of SANDY. The oracle version directly uses annotations from the DVQA dataset to build a DEM. The OCR version uses the output of the open-source Tesseract OCR. Tesseract's output is pre-processed in three ways: 1) we only use words with alphabetical characters in them, 2) we filter word detections with confidence less than 50%, and 3) we filter single-character word detections.

### 5.4.4 Training the Models

All of the classification based systems, except SANDY and the OCR branch of MOM, use a global answer dictionary from training set containing 1076 words, so they each have 1076 output units. MOM's OCR branch contains 27 output units; 1 for each alphabet and and 1 reserved for `blank` character. Similarly, SANDY's output layer contains 107 units,

Table 5.3: Overall results for models trained and tested on the DVQA dataset. Values are % of questions answered correctly.

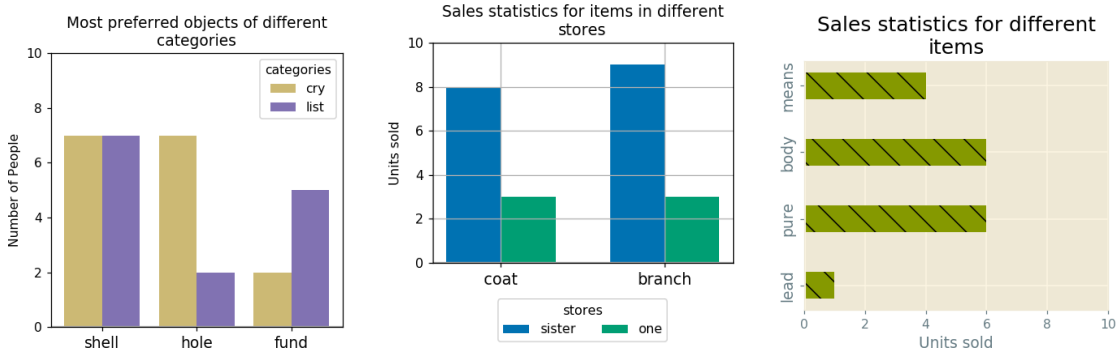
| | **Test-Familiar** | | | | **Test-Novel** | | | |
|---|---|---|---|---|---|---|---|---|
| | Structure | Data | Reasoning | **Overall** | Structure | Data | Reasoning | **Overall** |
| YES | 41.14 | 7.45 | 8.31 | 11.78 | 41.01 | 7.52 | 8.23 | 11.77 |
| IMG | 60.09 | 9.07 | 8.27 | 14.83 | 59.83 | 9.11 | 8.37 | 14.90 |
| QUES | 44.03 | 9.82 | 25.87 | 21.06 | 43.90 | 9.80 | 25.76 | 21.00 |
| IMG+QUES | 90.38 | 15.74 | 31.95 | 32.01 | 90.06 | 15.85 | 31.84 | 32.01 |
| SAN-VQA | 94.71 | 18.78 | 37.29 | 36.04 | 94.82 | 18.92 | 37.25 | 36.14 |
| MOM | 94.71 | 29.52 | 39.21 | 40.89 | 94.82 | 21.40 | 37.68 | 37.26 |
| MOM ($\pm$1) | 94.71 | 38.20 | 40.99 | 45.03 | 94.82 | 29.14 | 39.26 | 40.90 |
| SANDY (Oracle) | **96.47** | **65.40** | **44.03** | **56.48** | **96.42** | **65.55** | **44.09** | **56.62** |
| SANDY (OCR) | **96.47** | 37.82 | 41.50 | 45.77 | **96.42** | 37.78 | 41.49 | 45.81 |

Table 5.4: Results for chart-specific questions and answers. State-of-the-art VQA algorithms struggle with both.

| | **Test-Familiar** | | | **Test-Novel** | | |
|---|---|---|---|---|---|---|
| | Chart-specific Questions | Chart-specific Answers | **Overall** | Chart-specific Questions | Chart-specific Answers | **Overall** |
| YES | 17.71 | 0.00 | 11.78 | 17.58 | 0.00 | 11.77 |
| IMG | 17.61 | 0.00 | 14.83 | 17.88 | 0.00 | 14.90 |
| QUES | 23.17 | 0.10 | 21.06 | 22.97 | 0.00 | 21.00 |
| IMG+QUES | 25.52 | 0.09 | 32.01 | 25.49 | 0.00 | 32.01 |
| SAN-VQA | 26.54 | 0.10 | 36.04 | 26.32 | 0.00 | 36.14 |
| MOM | 26.54 | 12.78 | 40.89 | 26.32 | 2.93 | 37.26 |
| MOM ($\pm$1) | 26.54 | 23.62 | 45.03 | 26.32 | 12.47 | 40.90 |
| SANDY (Oracle) | **27.80** | **52.55** | **56.48** | **27.77** | **52.70** | **56.62** |
| SANDY (OCR) | 26.60 | 25.19 | 45.77 | 26.43 | 25.12 | 45.81 |

with the indices 31 through 107 are reserved for common answers and indices 0 through 30 are reserved for the local dictionary.

For a fair comparison, we use the same training hyperparameters for all the models and closely follow the architecture for SAN models from [124] wherever possible. SAN portion for all the models are trained using early stopping and regularized using dropout of 0.5 on inputs to all convolutional, fully-connected and LSTM units. All models use Adam [144] optimizer with an initial learning rate of 0.001.




**Q**: How many objects are preferred by less than 7 people in at least one category?
SAN: two ✓ MOM: two ✓ SANDY: two✓
**Q**: What category does the medium purple color represent?
SAN: closet ✗ MOM: lisit ✗ SANDY: list ✓

**Q**: Which item sold the most number of units summed across all the stores?
SAN: closet✗ MOM: branch✓ SANDY: branch✓
**Q**: How many units of the item branch were sold in the store sister?
SAN: 9 ✓ MOM: 9 ✓ SANDY: 9 ✓

**Q**: Are the values in the chart presented in a percentage scale?
SAN: no ✓ MOM: no ✓ SANDY: no ✓
**Q**: How many units of items lead and pure were sold?
SAN: 8 ✗ MOM: 8 ✗ SANDY: 7 ✓

Figure 5.5: Example results for different models on DVQA. Outputs of oracle version of SANDY model are shown. SAN completely fails to predict chart-specific answers whereas MOM model often makes small OCR errors (left). Both MOM and SAN are also incapable of properly encoding chart-specific labels in questions (right).

## 5.5 Experiments

In this section, we describe the experimental results for models trained and tested on the DVQA dataset. DVQA's extensive annotations are used to analyze the performance of each model on different question- and answer-types to reveal their respective strengths and weaknesses. In our experiments, we study the performance of algorithms on both familiar and novel chart labels, which are contained in two distinct test splits, Test-Familiar and Test-Novel. Every bar chart in Test-Familiar contains only labels seen during training. All of the models using the LSTM-encoder have entries in their word dictionaries for these familiar words, and all answers have been seen in the training set. The labels for the charts in Test-Novel are only seen in the test set, and no system has them in the dictionaries they use to encode words or to generate answers.

To measure performance, an algorithm gets a question correct only if it generates a string that is identical to the ground truth. To better assess MOM, we also measure its performance using edit distance, which is denoted MOM ($\pm 1$). This model is allowed to

get a question correct as long as the answer it generates is within one edit distance or less compared to the correct answer.

### 5.5.1 General Observations

Overall performance of each method broken down based on question-type are shown in Table 5.3 and some qualitative examples are shown in Fig 5.5. Across all question-types, NO, IMG, and QUES are the first, second, and third worst performing, respectively. Overall, SANDY performs best on both Test-Familiar and Test-Novel with SANDY-real following closely behind.

For structure questions, there is little difference across models for Test-Familiar and Test-Novel, which is expected because these questions ask about the general visual organization of a chart and do not require label reading. Performance increases greatly for IMG+QUES compared to either IMG or QUES, indicating structure questions demand combining image and question features.

For data retrieval and reasoning questions, SANDY and MOM both outperformed all baseline models. Both SANDY and SANDY-real outperformed MOM, and this gap was greater for Test-Novel.

### 5.5.2 Chart-specific Words in Questions and Answers

Many DVQA questions have chart-specific answers, *e.g.* , 'Which algorithm has the highest accuracy?' needs to be answered with the label of the bar with the highest value. These chart-specific answers are different than the generic answers that are shared across many bar charts, *e.g.* , 'Does the chart contain stacked bars?'. Similarly, some DVQA questions refer to elements that are specific to a given chart, *e.g.* , 'What is the accuracy of the algorithm **A**?'. To accurately answer these questions, an algorithm must be able to interpret the text-label **A** in the context of the given bar chart. Table 5.4 shows the accuracy of the algorithms for questions that have chart-specific labels in them (chart-specific questions) and questions whose answers contain chart-specific labels (chart-specific answers). As shown, whenever chart-specific labels appear in the answer, both IMG+QUES and SAN-VQA fail abysmally. While this is expected for Test-Novel, they perform no better on Test-Familiar. Likewise, all of the models except SANDY also face difficulty for questions with chart-specific labels. Overall, they fail to meaningfully outperform the QUES baseline. We believe that the small gain in accuracy by IMG+QUES and SAN-VQA over QUES is only because the image information, such as the type of scale used (linear, percentage, or logarithmic), enables these methods to guess answers with higher precision.

In chart-specific answers, SANDY showed highest accuracy. Moreover, its performance for Test-Novel is similar to that for Test-Familiar. In comparision, while MOM

outperforms the baselines, its accuracy on Test-Novel is much lower than its accuracy on Test-Familiar. This could be because MOM's string generation system is unable to produce accurate results with novel words. Supporting this, MOM often makes small string generation errors, as shown by the improved performance of MOM ($\pm 1$), which is evaluated using edit distance. MOM's output is also dependent on the precise prediction of the bounding box containing the answer which could further affect the final accuracy. MOM's localization performance is explored in more detail in the supplemental materials.

In addition to SANDY's ability to predict chart-specific answer tokens, it can also be used to properly tokenize the chart-specific words in questions. An LSTM based question encoder using a fixed vocabulary will not be able to encode the questions properly, especially when encoding questions with unknown words in Test-Novel. For questions with chart-specific labels on them, SANDY shows improvement in properly encoding the questions with the chart-specific labels compared to baselines. However, the improvement in performance is not as drastic as seen for chart-specific answers. This may be due to the fact that many of the chart-specific questions include precise measurement *e.g.* 'How many people prefer object **O**?' which could be beyond the capacity of the SAN architecture.

## 5.6 Discussion

In this chapter we presented the DVQA dataset and explored models for DVQA. Our experiments show that VQA algorithms are only capable of answering simple structure questions. They perform much more poorly on data retrieval and reasoning questions, whereas our approaches, SANDY and MOM, are able to better answer these questions. Moreover, SANDY and MOM can both produce answers that are novel to the test set, which is impossible for traditional VQA algorithms. Finally, SANDY can also encode questions with novel words from the bar chart.

We studied SANDY's performance using a real OCR and a perfect oracle OCR system. While the performance dropped when real OCR was used, it still surpassed other algorithms across all categories. Despite its success, the proposed dynamic encoding used in SANDY is simple and offers a lot of room for expansion. Currently, the dynamic encoding is inferred based on the position of previously detected words. Any error in the OCR system in detecting a single word will propagate throughout the chain and render the encoding for the whole image useless. While this is not a problem for a perfect OCR, removing the cascaded reliance on correctness of each OCR results can help improve performance for an imperfect real-world OCR system.

Recently, multiple compositional models for VQA, such as neural module networks (NMN) [82, 92, 122], have been developed. These recursive neural network systems consist of stacked sub-networks that are executed to answer questions, and they work well on

the compositional reasoning questions in CLEVR [83]. However, current NMN formulations are unable to produce chart-specific answers, so they cannot be used for DVQA without suitable modifications.

SANDY and MOM are both built on top of SAN, and they try to solve chart-specific answer generation in two distinct ways that are agnostic to SAN's actual architecture. SANDY uses DEM and OCR to encode an image's text, whereas MOM attempts to predict the location of the text it needs to generate an answer. As VQA systems continue to evolve, upgrading SAN with an better VQA algorithm could improve the performance of our systems.

Our dataset currently only contains bar charts. We are developing a follow-up version that will contain pie charts, plots, and other visualizations in addition to bar-charts. Since neither MOM nor SANDY are designed specifically for bar-charts, they can operate on these alternative diagrams with only minor modifications.

We conducted an additional study to assess how well these models work on real bar charts. We manually annotated over 500 structure understanding questions for real bar charts scraped from the Internet. Without any fine-tuning, all of the SAN-based models achieved about 59% accuracy on these questions, a 15% absolute improvement over the image blind (QUES) baseline. This shows a positive transfer from synthetic to real-world bar charts. Training on entirely real charts would be ideal, but even then there would likely be a benefit to using synthetic datasets as a form of data augmentation [145].

## 5.7 Conclusion

Here, we described DVQA, a dataset for understanding bar charts. We demonstrated that VQA algorithms are incapable of answering simple DVQA questions. We proposed two DVQA algorithms that can handle chart-specific words in questions and answers. Solving DVQA will enable systems that can be utilized to intelligently query massive repositories of human-generated data, which would be an enormous aid to scientists and businesses. Automatically parsing bar-charts can also enable visually impaired users to fully parse contents of a document. We hope that the DVQA dataset, which will be made publicly available, will promote the study of issues that are generally ignored with VQA with natural images, *e.g.* , out-of-vocabulary words and dynamic question encoding. We also hope that DVQA will serve as an important proxy task for studying visual attention, memory, and reasoning capabilities.

# Chapter 6

# Answering Questions about Data Visualizations using Efficient Bimodal Fusion

## 6.1 Introduction

As discussed in preceding chapter, chart QA (CQA) is a VQA task involving answering questions about data visualizations. Formally, given an data visualization image $I$ and a question $Q$ about $I$, a CQA model must predict the answer $A$. CQA requires understanding of the relationships among different 'symbols' (elements in the chart) in an image. In contrast to natural images, even tiny modifications to the image can cause drastic changes in the correct answer, making CQA an excellent platform for studying reasoning mechanisms as described in chapter 5. Concurrent with DVQA, introduced in chapter 5, another dataset for answering questions about data visualizations were introduced along with a few new algorithms [2]; however, there is considerable room for improvement. Here, we propose a novel algorithm that exceeds the state-of-the-art on both of these datasets by a large margin.

In this chapter, we describe a novel algorithm called parallel recurrent fusion of image and language (PReFIL). PReFIL jointly learns bimodal embeddings by using both low- and high-level image features, which enable it to answer complex questions requiring multi-step reasoning and comparison without employing specialized relational or attention modules. Extensive experiments show that our algorithm outperforms current state-of-the-art methods, by a large margin in two challenging CQA datasets.

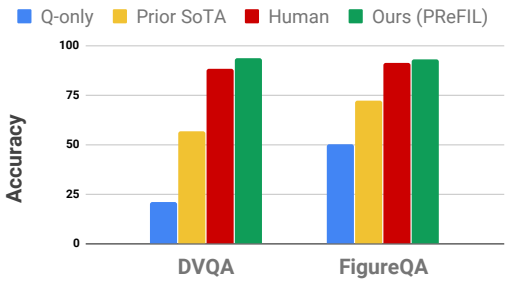



Figure 6.1: We propose the PReFIL algorithm for chart question answering (CQA). PReFIL surpasses the prior state-of-the-art (SoTA) and human baselines on DVQA and FigureQA datasets.

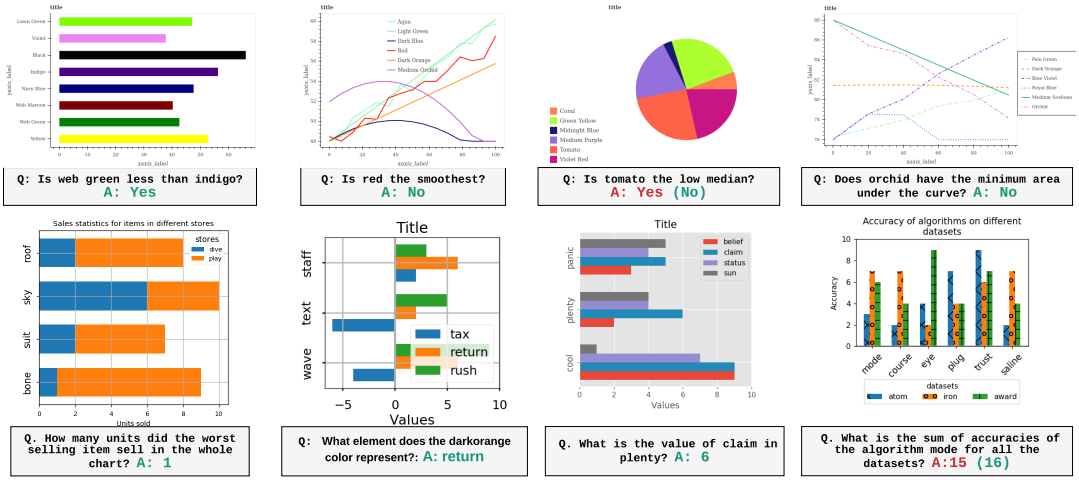



Figure 6.2: Example images and PReFIL outputs for FigureQA (top) and DVQA (bottom). Red denotes incorrect predictions. For incorrect predictions, correct answer is shown in parentheses. More examples are included in the supplementary materials.

**Key issues addressed in this chapter are:**

- We propose a novel algorithm called parallel recurrent early fusion of image and language (PReFIL) (Sec. 6.2). PReFIL greatly surpasses existing methods on CQA datasets and also outperforms humans on both DVQA and FigureQA (Sec. 6.3). PReFIL's code and pre-trained models will be publicly released.
- We collect human performance values for the DVQA dataset using crowd-sourcing (Sec. 6.3).
- We pioneer the use of iterative question answering to reconstruct tables from charts (Sec. 6.3.4).
- In light of our results, we outline a road map toward creating more challenging

Table 6.1: FigureQA vs. DVQA

| | Images | QA Pairs | Ques. Format | Chart Types | OCR | OOV |
|---|---|---|---|---|---|---|
| DVQA | 300,000 | 3,487,194 | Open-ended | 1 | Yes | Yes |
| FigureQA | 180,000 | 2,38,8698 | Yes/No | 5 | No | No |

datasets and algorithms for understanding data visualizations (Sec. 6.4).

### 6.1.1 Datasets for CQA

Two CQA datasets: DVQA (chapter 5) and FigureQA [2], are publicly available at the time of writing this chapter. See Table 6.1 for their statistics. Example images are shown in Fig. 6.2. Since chapter 5 mostly focused construction of the DVQA dataset and creation of baseline algorithms, we did not discuss FigureQA fully. Here, we will describe and compare FigureQA and DVQA in more detail.

**DVQA** has over 3 million question answer pairs for 300,000 images for bar charts. The question answer pairs in DVQA are divided into three categories: 1) structure understanding (e.g. "How many bars are there?"), 2) data query (e.g., "How many units of item $X$ were sold?"), and 3) reasoning (e.g. "Is the accuracy of algorithm $X$ greater than algorithm $Y$?"). Since many questions refer to texts specific to the corresponding charts, systems must integrate OCR and dynamically expand their vocabulary to correctly answer questions. DVQA has two test splits: Test-Familiar and Test-Novel, with Test-Novel containing charts with texts that were not seen during training.

**FigureQA** has over 2 million question answer pairs for 180,000 images. It has five kinds of visualizations: 1) vertical bar charts, 2) horizontal bar charts, 3) pie charts, 4) line graphs and 5) dot-line graphs. Chart element colors are uniformly distributed in the training and validation sets. FigureQA has harder versions of the validation and test sets with color combinations that are unseen in the training set. Validation 1 and Test 1 have the same colors as the training set and Validation 2 and Test 2 have a color scheme that differs from training. Test set annotations are not publicly available. All questions are binary (yes/no) and demand multiple abilities, including finding the largest/smallest element (e.g. "Is $X$ the largest/smallest?"), comparing values of two elements (e.g. "Is $X$ greater/smaller than $Y$?"), and other scientific measurements (e.g. "Does $X$ have maximum area under the curve?").

#### DVQA versus FigureQA

DVQA and FigureQA each have their own strengths and shortcomings. We compare and contrast them below.

**Shared strengths:** Both datasets are large and provide enough training samples to train large scale models, e.g. in DVQA, each unique visual element is repeated at least 1,000 times. Both datasets provide detailed annotations for all figure elements in addition to the question answer pairs, making it possible to create auxiliary tasks or use them as additional training signals. The creators of both datasets tried to eliminate some sources of bias. DVQA has randomized visual elements and it also has a balanced question answer distribution to make guessing difficult. Similarly, FigureQA has a randomized distribution of colors and a balanced distribution of "yes" and "no" answers for each unique question template. Lastly, both datasets provide both easy and hard test splits, where the hard test split measures generalization beyond what is seen during training. DVQA's "Test Novel" split measures generalization to unseen words and FigureQA provides an "alternated colors" split where visual elements in the chart have different colors than the ones seen during training.

**DVQA's advantages:** In DVQA, questions about bars are asked by referring to their text labels, e.g. "What is the value of algorithm $X$?" where $X$ is an actual label in the chart and it will be different for each chart even if they have the same appearance, e.g. identical red bars may have label $X$ in one image and $Y$ in another. This requires integrating OCR into the system. In contrast, FigureQA refers to chart elements by their color, e.g. "red bars" will always be referred to as "red" making it easier for systems to identify a chart's elements. Since DVQA uses chart labels, algorithms must take into account that some of the words may be out-of-vocabulary (OOV) and unseen during training for both questions and answer. To handle this, systems need to have a vocabulary that can be dynamically adjusted during testing. FigureQA has no OOV answers. DVQA also tests for more tasks than FigureQA. For bar charts, DVQA contain most of the tasks in FigureQA (e.g. identifying colors, comparing values, etc.) and several that are not required for FigureQA (e.g. data measurement and inferring structure of the chart). Finally, while DVQA contains only bar charts, its bar charts have increased visual complexity compared to those in FigureQA. FigureQA is limited to single-variable vertical and horizontal bar charts, whereas DVQA also has grouped bar charts and stacked bar charts with legends. DVQA's bars can be hatched, monochrome, and have negative values, all of which are absent in FigureQA.

**FigureQA's advantages:** While DVQA has only bar charts, FigureQA has three kinds of data visualizations: bar charts, pie charts, and line graphs. This allows FigureQA to have unique question-types that are not encountered for bar chart alone. E.g., for line graphs, FigureQA requires determining the area under the curve, and whether one line intersects another. These are not tested in DVQA. FigureQA also tests compositional reasoning by asking questions about unknown color combinations in chart elements, whereas colors are randomly distributed in DVQA.

**Shared limitations:** As synthetically generated datasets, both DVQA and FigureQA omit much of the variability found in real-world data visualizations. All of DVQA's charts

were made with Matplotlib and all of FigureQA's were made with Bokeh. The variation introduced is limited to the capabilities of these packages. FigureQA uses only generic titles and other chart elements. DVQA has some variety but ultimately is limited to a few templates. Likewise, both datasets have formulaic, templated questions. While questions can be complex, they lack the diversity of human generated queries. In the discussion we elaborate further on how future datasets could overcome these limitations.

### 6.1.2 Existing CQA Algorithms

For DVQA, we discussed SANDY (SAN with DYnamic encoding), which is a modified version of the stacked attention network (SAN) [91, 124]. SAN cannot handle DVQA's OOV words in its test set or the chart specific words found in its questions and answers. To address this, SANDY uses an off-the-shelf OCR method to recognize such words and introduced dynamic encoding to represent OOV and chart-specific words. SANDY's dynamic encoding scheme for OCR can be incorporated into any classification-based VQA algorithm.

FigureQA's creators used a relation network (RN) [1] on their dataset. RN encodes pairwise interactions between every pair of "objects" in an image, enabling it to answer questions involving relationships. Each "object" is a cell of a convolutional feature map. RN has been shown to be especially effective at compositional reasoning in CLEVR [1], and it exceeded baselines on FigureQA.

FigureNet [146] is a multi-step algorithm for FigureQA composed of different modules. The first module is called the spectral segregator, which identifies the elements and colors of the chart. It is followed by the extraction module, which quantifies the values represented by each element. This is then used with a feed-forward network to predict the answer. FigureNet uses the detailed annotations of FigureQA's chart elements to pre-train each of the modules. Because FigureNet relies on having access to the measurements of each chart element, they could only apply it to FigureNet's bar and pie charts.

To assess bias in their datasets, both FigureQA and DVQA studies question-blind and image-blind models. These models performed abysmally indicating that vision and language must be jointly used to correctly answer the questions. Both datasets also tested simple question+image fusion schemes. These worked better than the blind baselines, but this did not suffice for handling the complexity found in CQA. This is in contrast to VQA with natural images, where these algorithms fare comparatively well.

Compared to prior existing work, our model does not employ complex attention or relational modules, and unlike FigureNet, it does not require additional supervised annotations for training on FigureQA.

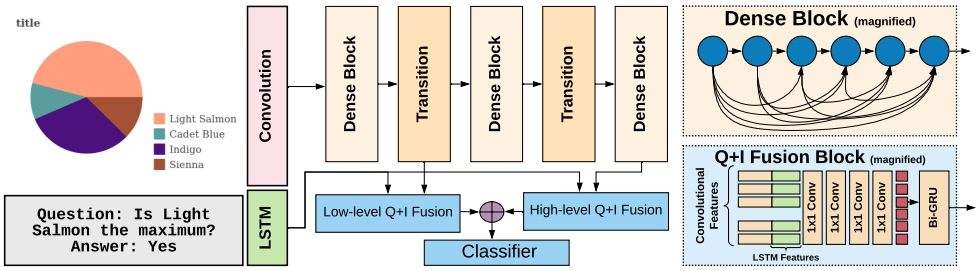



Figure 6.3: Components of our PReFIL model. Magnified views show the details of each dense block and Q+I fusion block.

## 6.2 The PReFIL Model

We propose the PReFIL algorithm for CQA. As shown in Fig. 6.3, PReFIL has two parallel Q+I fusion branches. Each branch takes in question features (from an LSTM) and image features from two locations of a 40-layer DenseNet, *i.e.* low-level features (from layer 14) and high-level features (from layer 40). Each Q+I fusion block concatenates the question features to each element of the convolutional feature map, and then it has a series of $1 \times 1$ convolutions to create question-specific bimodal embeddings. These embeddings are recurrently aggregated and then fed to a classifier that predicts the answer. Despite being composed of relatively simple elements, PReFIL outperforms more complex methods that use RNs and attention mechanisms. The three main stages of PReFIL are described in the next subsections. For DVQA, an additional fourth OCR-integration component is required (Sec. 6.2.4). In Sec. 6.3.3, we conduct studies to understand the value of each stage.

### 6.2.1 Multi-stage Image Encoder

For all model variants, image encoder is a DenseNet [147] trained from scratch. DenseNet is an efficient architecture for training deep convolutional neural networks (CNNs). It is comprised of several "dense blocks" and "transition blocks" between the dense blocks. Each dense block has several convolutional layers, where each layer uses outputs of all preceding layers as its input. The transition block sits between two dense blocks and serves to change feature-map sizes via convolution and pooling. This architecture encourages feature reuse, improves training, and mitigates vanishing-gradients, making it easy to train very deep networks. Feature reuse allows DenseNet to learn complex visual features with fewer parameters compared to other architectures [148].

In deep CNNs, complex features are learned as a hierarchy of visual features with earlier layers learning simple features and later layers learning higher-level features that are combinations of simpler features [149]. In data visualizations, simpler features such

as color patches, lines, textures, etc. convey important information that is often abstracted away by deeper layers of a CNN. Hence, we use both low- and high-level convolutional features in our model, both of which are fed to parallel fusion module alongside question embeddings learned using an LSTM. We study the importance of both low and high level features in Sec. 6.3.3.

### 6.2.2 Parallel Fusion of Image and Language

Jointly modulating visual features using vision and language features can allow models to learn richer features for downstream tasks [150–152]. Our Q+I fusion block does this by first concatenating all of the input convolutional feature map's spatial locations with the question features, and then bimodal fusion occurs using a series of layers that use $1 \times 1$ convolutions [150, 152]. This allows the question to modulate visual feature processing and yields bimodal embeddings that capture information from both the image and the question. This approach resembles early VQA models that concatenated CNN embeddings to question embeddings, with the critical difference being that this happens before spatial pooling across the entire scene. We do this for both low-level and high-level convolutional features in parallel. In Sec. 6.3.3, we study the importance of learning bimodal embeddings jointly.

### 6.2.3 Recurrent Aggregation of bi-modal features

In CNNs, the most common approach to aggregating information from a feature map $F \in \mathbb{R}^{M \times N \times D}$ is to collapse across the spatial dimensions to produce a $D$ dimensional vector by mean pooling or max pooling. An alternative is to "flatten" $F$ to turn it into a $DMN$-dimensional vector. Recent attentive approaches have explored using a weighted sum, where the relative importance of each region is based on the question. These methods may fail to capture *interactions* among features, especially for high-level tasks such as question answering. To address this, we aggregate information using a bidirectional gated recurrent unit (bi-GRU), which sequentially takes in the $D$-dimensional features from each of the $MN$ locations in $F$. The aggregated features are sent to a classifier to predict the answer. As ablation, we also try sum-pooling for aggregation in Sec. 6.3.3.

### 6.2.4 OCR Integration for DVQA dataset

Unlike FigureQA and most VQA tasks, DVQA requires OCR to answer its reasoning and data questions. A fixed vocabulary consisting of all the words seen during training is not enough since the model will encounter OOV words during testing. To integrate OCR into PReFIL, we use the same dynamic encoding scheme used by the SANDY model.

To assess impact of OCR, we test three OCR versions as well as a version of algorithm trained without the dynamic encoding, i.e., only using a fixed-vocabulary constructed from the train split. The first two OCR systems are identical to those used in chapter 5: an oracle (perfect) OCR model and a real OCR system using Tesseract. Because Tesseract has been found to be sub-optimal when used directly on diagrams [141], we also study using a two-stage OCR pipeline where we first detect text and then run OCR on the detected regions to recognize the text. Specifically, we use the EAST text detector [153] to detect text-regions for images rotated at 0, 45 and 90 degrees. We then perform non-maximum suppression on overlapping detections and crop them. Each cropped region is resized by 200% and sent to the Tesseract OCR to obtain the text within each region. The rest of the dynamic encoding scheme remains unchanged.

### 6.2.5 Model and Training Hyperparameters

**Question Encoding:** Question words are represented by 32 dimensional learned word embedding and passed through an LSTM which provides a 256-dimensional embedding representing the whole question.

**DenseNet:** We use a 40 layer DenseNet composed of 3 dense blocks with 12 layers each. The number of initial filters is 64 and the growth rate is set to 32.

**Preprocessing:** DVQA images are resized to a size of $256 \times 256$. FigureQA images are all differently sized but we resize them to $320 \times 224$ which maintains an *average* width-height aspect ratio. For data augmentation during training, both DVQA and FigureQA images are padded with 8 pixels on all sides, followed by random crops and random rotations of up to 3 degrees.

**Q+I Fusion:** Inputs to Q+I block are batchnormed. Each Q+I fusion block is composed of four $1 \times 1$ convolutions with 256 channels and ReLU.

**Recurrent Fusion:** The bimodal features are aggregated using a 256 dimensional bi-directional GRU. The forward and backward direction outputs are concatenated to form a 512 dimensional vector which is fed to the classifier.

**Classifier:** The aggregated bimodal features are projected to a 1024 fully connected ReLU layer, which was regularized using dropout of 0.5 during training. The classification layer is binary for FigureQA. For DVQA, the classification layer has 107 units, with 77 units for predicting 'common' answers such as 'yes', 'no', 'three groups', etc, and 30 special tokens for predicting answers that require OCR, which allows PReFIL to produce OOV answer tokens that are unseen during training (see Sec. 6.2.4 for details).

**Losses and Optimizers:** For DVQA, PReFIL is trained using multinomial cross-entropy loss. For FigureQA, PReFIL is trained using binary cross entropy loss. Following [104], we use Adamax optimizer with a gradual learning rate warm-up, with a base learning rate of $7 \times 10^{-4}$. The first 4 epochs use a learning rate of $(0.5 \times epoch \times base)$ and the rate starts decaying by a factor of 0.7 from epochs 15 to 25. For DVQA, all models

are trained for a fixed 25 epochs. For FigureQA, we train them until they converge on the validation set and submit predictions to its creators for assessment on the non-public test set.

## 6.3 Experiments and Results

### 6.3.1 FigureQA

FigureQA has two validation sets and two non-publicly available test sets. Validation 1 and Test 1 have the same colors as the training set and Validation 2 and Test 2 have a color scheme that differs from training. Test sets are not publicly available and the results were obtained by sending the predictions to the authors. Existing works do not report accuracy for the full test set, but we report results for both validation and test sets in Table 6.2 for completeness.

Our PReFIL algorithm exceeds FigureNet by a large margin despite FigureNet having access to additional annotations. FigureNet is incapable of answering questions about line and dot-line graphs, so it is only evaluated on vBar, hBar and Pie. For these chart types, average accuracy for FigureNet is 83.9%, compared to 97.33% for ours.

FigureQA also provides human performance for a *subset* of Test 2, which is not available for the other sets. We report PReFIL's performance compared to other baselines and human performance on the exact same subset in Table 6.3. PReFIL outperforms the human baseline for four out of five categories and also surpasses overall human accuracy. When analyzed for different question templates, PReFIL outperforms humans for 12 out of 15 question templates. PReFIL shows the most improvements for questions requiring measurements, e.g. for the question template "Is X the high/low median?" PReFIL outperforms human accuracy by over 7% (absolute). Detailed results for all 15 templates are presented in the supplementary materials.

### 6.3.2 DVQA

DVQA is split into Test-Familiar, which contains bar charts with words that are also encountered in its Train set, and Test-Novel, which contains bar charts with novel words in them. Results for both DVQA splits are given in Table 6.4. PReFIL surpasses SANDY by over 40% in accuracy when both the baseline SANDY and our PReFIL method have access to a perfect Oracle OCR, which is emulated by providing the correct text-annotations for all the elements in the images. When using Tesseract OCR, we obtain about a 24% improvement overall on both test sets. To demonstrate that PReFIL's performance scales with access to better OCR, we also test a version that uses an improved OCR pipeline (see Sec. 6.2). This further improves PReFIL's performance by about 11% bringing it closer to

Table 6.2: Results for the FigureQA dataset for our PReFIL algorithm compared to baseline and existing algorithms.

| | **Validation 1 - Same Colors** | | | | | | **Validation 2 - Alternated Colors** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | vBar | hBar | Pie | Line | Dot-line | Overall | vBar | hBar | Pie | Line | Dot-line | Overall |
| QUES [2] | - | - | - | - | - | - | - | - | - | - | - | 50.01 |
| IMG+QUES [2] | 61.98 | 62.44 | 59.63 | 57.07 | 57.35 | 59.41 | 58.60 | 58.05 | 55.97 | 56.37 | 56.97 | 57.14 |
| RN [2] | 85.71 | 80.60 | 82.56 | 69.53 | 68.51 | 76.39 | 77.35 | 77.00 | 74.16 | 67.90 | 69.04 | 72.54 |
| FigureNet [146] | 87.36 | 81.57 | 83.13 | - | - | - | - | - | - | - | - | - |
| PReFIL (Ours) | **98.80** | **98.09** | **95.11** | **91.82** | **92.19** | **94.84** | **98.46** | **97.94** | **93.57** | 88.50 | **90.30** | **93.26** |
| | **Test 1 - Same Colors** | | | | | | **Test 2 - Alternated Colors** | | | | | |
| PReFIL (Ours) | 98.79 | 98.14 | 95.35 | 91.98 | 92.05 | 94.88 | 98.41 | 97.93 | 93.58 | 88.26 | 90.07 | 93.16 |

Table 6.3: Results on FigureQA's Test 2 split with alternated color schemes. All results are from the 16,876 questions answered by human annotators.

| Type | PReFIL(Ours) | Q+I [2] | RN [2] | Human [2] |
|---|---|---|---|---|
| vBar | **98.25** | 59.63 | 77.13 | 95.90 |
| hBar | **97.98** | 57.69 | 77.02 | 96.03 |
| Pie | **92.84** | 55.32 | 73.26 | 88.26 |
| Line | 87.79 | 54.46 | 66.69 | **90.55** |
| Dot-line | **89.57** | 54.19 | 69.22 | 87.20 |
| Overall | **92.79** | 56.04 | 72.18 | 91.21 |

the results of the oracle OCR version. When OCR is removed entirely, PReFIL still performs about 11% better than SANDY without OCR, but this ablation renders many data and reasoning questions impossible to answer. This re-affirms the assertion by DVQA's creators that OCR integration is essential for answering the data and reasoning questions in the dataset [31].

Across all OCR variants, PReFIL outperforms SANDY. Moreover, PReFIL's performance scales much better when better OCR is available: 11% gain for SANDY vs. 26% gain for PReFIL when moving from the imperfect Tesseract OCR setup to the perfect Oracle OCR setup. Our results show that PReFIL is as effective for novel words (Test-Novel) as it is for familiar words (Test-Familiar). This is enabled by the dynamic OCR integration, which is designed to be agnostic to whether a word has been encountered before.

Because no human accuracy estimate for DVQA existed, we had people answer 5000 randomly selected questions for 5000 images from the DVQA Test-Novel split. The annotators were shown example QA pairs from each of three question types. We perform post processing on the provided answers to rectify minor answer entry errors. First, we found some annotators used decimal points or spelled out numerals ("5.0" or "five" instead of "5") despite our instructions to only use integers when answers are numbers. Because DVQA contains only integers, we convert all such occurrences to the nearest integer. For word answers, we allow one character typographic error to be discounted. Results for humans and models are given in Table 6.4. With perfect OCR, PReFIL surpasses the DVQA human accuracy result across question types. Its performance on reasoning questions is almost 10% greater (absolute), and it exceeds them by almost 8% (absolute) for DVQA's data questions, which require measurement. However, without perfect OCR humans exceed PReFIL, although the better OCR used for PReFIL does lead to significantly better results than PReFIL with improved OCR. This suggests that the underlying core algorithm and reasoning mechanisms in PReFIL work well for DVQA, and the main limiting factor is OCR.

Table 6.4: Results for the DVQA dataset for PReFIL compared to baselines and existing algorithms.

| | **Test-Familiar** | | | | **Test-Novel** | | | |
|---|---|---|---|---|---|---|---|---|
| | Structure | Data | Reasoning | Overall | Structure | Data | Reasoning | Overall |
| QUES [31] | 44.03 | 9.82 | 25.87 | 21.06 | 43.90 | 9.80 | 25.76 | 21.00 |
| IMG+QUES [31] | 90.38 | 15.74 | 31.95 | 32.01 | 90.06 | 15.85 | 31.84 | 32.01 |
| SANDY (No OCR) [31] | 94.71 | 18.78 | 37.29 | 36.02 | 94.82 | 18.92 | 37.25 | 36.14 |
| PReFIL (No OCR) | 99.77 | 23.39 | 49.05 | 47.70 | 99.77 | 23.43 | 49.21 | 47.86 |
| SANDY (Tesseract OCR) [31] | 96.47 | 37.82 | 41.50 | 45.77 | 96.42 | 37.78 | 41.49 | 45.81 |
| PReFIL (Ours, Tesseract OCR) | 99.75 | 49.00 | 74.61 | 69.63 | 99.73 | 48.91 | 74.07 | 69.53 |
| PReFIL (Ours, Improved OCR) | 99.73 | 68.55 | 83.44 | 80.88 | 99.57 | 67.13 | 80.73 | 80.04 |
| SANDY (Oracle OCR) [31] | 96.47 | 65.40 | 44.03 | 56.48 | 96.42 | 65.55 | 44.09 | 56.62 |
| PReFIL (Ours, Oracle OCR) | **99.77** | **95.80** | **95.86** | **96.37** | **99.78** | **96.07** | **95.99** | **96.53** |
| Human | - | - | - | - | 96.19 | 88.70 | 85.83 | 88.18 |

Table 6.5: PReFIL ablation studies on a 500K DVQA train subset.

| Ablation Model | Test Familiar | Test Novel |
|---|---|---|
| PReFIL (full model) | **91.18** | **91.32** |
| No bimodal embedding | 78.00 | 78.36 |
| No high-level features | 85.68 | 85.86 |
| No low-level features | 89.87 | 90.05 |
| No recurrent aggregation | 90.88 | 91.14 |

### 6.3.3 Ablation Studies

We studied the contribution of PReFIL's components by analyzing a series of ablation models. We trained each model variation and the original PReFIL (Oracle OCR) for 25 epochs on a subset of DVQA that has only 500,000 randomly selected training samples. The ablation models are:

- **No bimodal embeddings:** Instead of learning bimodal embeddings, the question is concatenated after the recurrent aggregation and fed to the classifier.
- **No low-level features:** Only the high-level (layer 40 output) DenseNet features are used.
- **No high-level features:** Only the low-level (layer 14 output) DenseNet features are used. This is equivalent to using a shallower DenseNet.
- **No recurrent aggregation:** Instead of recurrent aggregation, output is aggregated via summation.

As shown in Table 6.5, all of PReFIL's components impact its performance. Removing bimodal embeddings causes the largest accuracy drop (over 12% absolute). The next largest is caused by removing low and high-level visual features (1.3% and 6% absolute).

### 6.3.4 Table Reconstruction by Asking Questions

We introduce table reconstruction for DVQA as an application of PReFIL. DVQA's question templates provide the questions needed to completely reconstruct its bar charts by iteratively asking questions about each chart. Our approach is given in Algorithm 1. An example reconstruction is shown in Fig. 6.4, and results using PReFIL (Oracle OCR) are given in Table 6.6. Shape prediction can be done with near perfect accuracy, but there is a drop in performance for both label and value prediction. To study the accuracy of different components in chart reconstruction, we also report accuracy on three main components of the iterative question-answering: 1) Shape prediction: Questions about number of bars and legends in the picture; 2) Label prediction: Predicting the label of given bar or legend; and 3) Value Prediction: Predicting the value of a given bar.

**Algorithm 1:** Iterative QA for Data Reconstruction

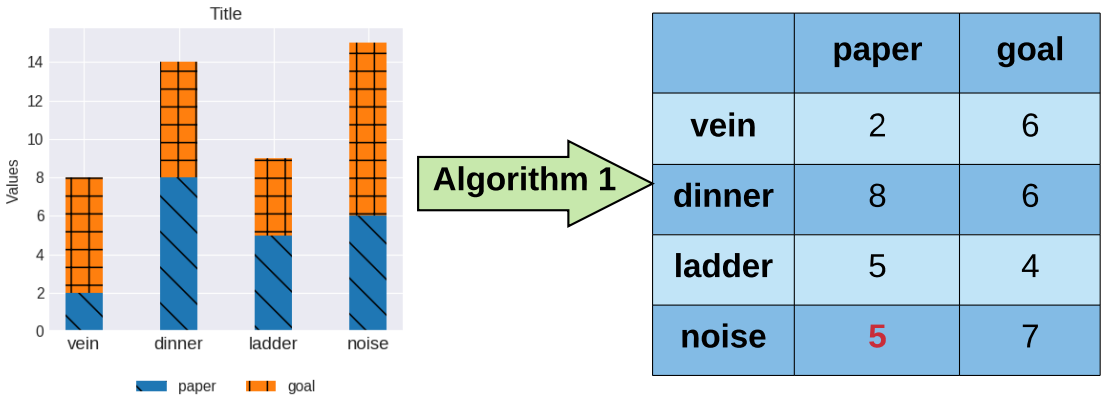
```
if bar_type is single then
    n = ans("How many bars are there?");
    for i ← 1 to n do
        data[i] = ans("What is the value of the i^th bar?");
        label[i] = ans("What is the label of the i^th bar?");
else
    m = ans("How many groups are there?");
    n = ans("How many bars are there per group?");
    for j ← 1 to n do
        legend_label[j] = ans("What is the label of the j^th bar in each group?");
    for i ← 1 to m do
        bar_label[i] = ans("What is the label of the i^th group?");
        for j ← 1 to n do
            data[i,j] = ans("What is the value of the j^th bar in i^th group?");
```

Table 6.6: Bar chart reconstruction accuracy (%) using Algorithm 1 with PreFIL (Oracle OCR).

| | Test Familiar | Test Novel |
|---|---|---|
| Shape Prediction | 99.97 | 99.97 |
| Label Prediction | 97.78 | 97.78 |
| Value Prediction | 84.21 | 84.75 |
| Overall | 90.79 | 91.10 |

## 6.4 Conclusion

We proposed PReFIL, a new CQA system that surpassed prior state-of-the-art methods for both DVQA and FigureQA. While PReFIL exceeded the human baseline for FigureQA, results are more nuanced for DVQA due to OCR model variations. All OCR versions exceeded the human baseline for structure questions, but only PReFIL using oracle OCR exceeded humans across all question types. We found that better OCR methods led to better results for DVQA. Future developments in OCR technology would likely improve PReFIL further. Our work has the potential to improve retrieval of information from charts, which has numerous applications, including automatic information retrieval, table reconstruction, and enabling better understanding of charts by people with visual impairments. The strong results in this chapter suggest that the community is ready for more difficult CQA datasets. We will discuss some future research directions in chapter 8.




| | paper | goal |
|---|---|---|
| vein | 2 | 6 |
| dinner | 8 | 6 |
| ladder | 5 | 4 |
| noise | 5 | 7 |

Figure 6.4: An example output of the chart to table algorithm. Red denotes incorrect predictions.

# Chapter 7

# Challenges in Vision and Language Research

## 7.1 Introduction

The primary objective of many scientists working on V&L problems is to have them serve as stepping stones toward a visual Turing test [40], a benchmark for progress in artificial intelligence (AI). To pass the visual Turing test, a V&L algorithm must demonstrate a robust understanding of natural language and an ability to visually ground the linguistic concepts in the form of objects, their attributes and their relationships. In earlier chapters, we discussed several issues with VQA and discussed several potential solutions. However, challenges in V&L language are neither unique to VQA, nor completely addressed. In this chapter, we will outline several challenges facing several V&L tasks, including VQA, that are yet unresolved.

As mentioned earlier, integrating vision and language provides a test-bed for assessing both natural language understanding and goal-directed visual understanding. V&L tasks can demand many disparate computer vision and NLP skills to be used simultaneously. For example, the same system may be required to simultaneously engage in entity extraction, entailment and co-reference resolution, visual and linguistic reasoning, object recognition, attribute detection, and much more. Most V&L benchmarks capture only a fraction of the requirements of a rigorous Turing test; however, we argue that a rigorous evaluation should test each capability required for visual and linguistic understanding *independently*, which will help in assessing if an algorithm is right for the right reasons. If it is possible to do well on a benchmark by ignoring visual and/or linguistic inputs, or by merely guessing based on spurious correlations, then it will not satisfy these requisites for a good test.

In this chapter, we identify the challenges in developing good algorithms, datasets, and evaluation metrics. We discuss issues unique to individual tasks as well as identify com-

mon shortcomings shared across V&L benchmarks. Finally, we provide our perspective on potential future directions for V&L research. In particular, we argue that both content and evaluation procedure of future V&L benchmarks should be carefully designed to mitigate dataset bias and superficial correlations. To this end, we propose a few concrete steps for the design of future V&L tasks that will serve as robust benchmarks for measuring progress in natural language understanding, computer vision, and the intersection of the two.

## 7.2 Shortcomings of V&L research

Progress in V&L research appears to be swift. For several V&L benchmarks, algorithms now rival human performance [57, 84]. However, these results are misleading because they ensue from the shortcomings in benchmarks rather than an algorithm's capability of true V&L understanding. In this section, we describe several such shortcomings.

### 7.2.1 Dataset bias

Dataset bias is a serious challenge faced by both computer vision [154, 155] and NLP [22, 156] systems. Because V&L systems operate at the intersection of the two, unwanted and unchecked biases are very prevalent in V&L tasks too. Since the data used for training and testing a model are often collected homogeneously [12, 34, 80], they share common patterns and regularities. Hence, it is possible for an algorithm to get good results by memorizing those patterns, undermining our efforts to evaluate the understanding of vision and language. The biases in datasets can stem from several sources, can be hard to track, and can result in severely misleading model evaluation. Two of the most common forms of bias stem from bias in crowd-sourced annotators and naturally occurring regularities. Finally, 'photographer's bias' is also prevalent in V&L benchmarks, because images found on the web share similarities in posture and composition due to humans having preferences for specific views [157]. Since the same biases and patterns are also mirrored in the test dataset, algorithms can simply memorize these superficial patterns (If the question has the pattern 'Is there an `OBJECT` in the picture?', then answer 'yes') instead of learning to actually solve the intended task (answer 'yes' only if the `OBJECT` is actually present). If this bias is not compensated for during evaluation, benchmarks may only test a very narrow subset of capabilities. This can enable algorithms to perform well for the wrong reasons and algorithms can end up catastrophically failing in uncommon scenarios [16, 158].

Several studies demonstrate the issue of bias in V&L tasks. For example, blind VQA models that 'guess' the answers without looking at images show relatively high accuracy (3). In captioning, simple nearest neighbor-based approaches yield surprisingly good

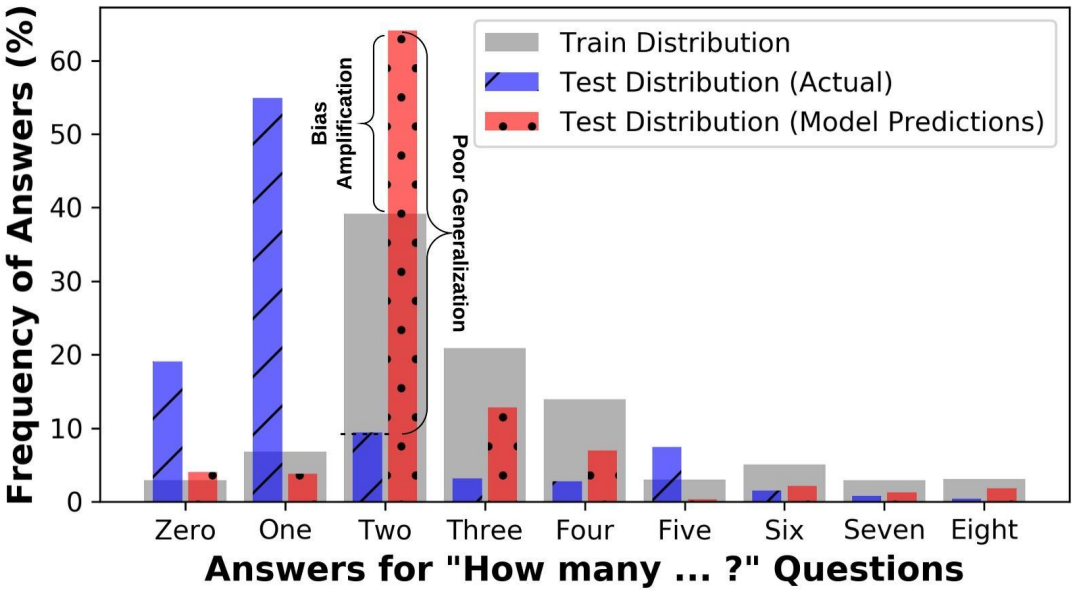



Figure 7.1: Answer distribution for questions starting with the phrase 'How many' in the train and test splits of VQA-CP dataset [16], alongside the test-set predictions from a state-of-the-art VQA model, BAN [104]. In VQA-CP, the distribution of test set is intentionally made different from the training set to assess if the algorithms can perform well under changing priors. Algorithms not only fail to perform well under changing priors, but they also demonstrate bias-amplification, i.e., the predictions show increased bias towards answers that are more common in the training set than the actual level of bias.

results [58]. Dataset bias occurs in other V&L tasks as well [22,26,64,159]. Recent studies [22] have shown that algorithms not only *mirror* the dataset bias in their predictions, but in fact *amplify* the effects of bias (see Fig. 7.1).

Numerous studies have sought to quantify and mitigate the effects of answer distribution bias on an algorithm's performance. As a straightforward solution, [14] and one of our earlier work (Chapter 4) proposed balanced training sets with a uniform distribution over possible answers. This is somewhat effective for simple binary questions and synthetically generated visual scenes, but it does not address the imbalance in the kinds of questions present in the datasets. Re-balancing all kinds of query types is infeasible for large-scale natural image datasets. Furthermore, it may be counterproductive to forgo information contained in natural distributions in the visual and linguistic content, and focus should instead be on rigorous evaluation that compensates for bias or demonstrates bias robustness [16]. We discuss this further in the next section.

### 7.2.2 Evaluation metrics

Proper evaluation of V&L algorithms is difficult. In computer vision, challenges in evaluation can primarily be attributed to class imbalance and dataset bias [160, 161]. Evaluation of NLP algorithms often poses greater challenges since the notion of *goodness* is ill-defined for natural language. These challenges, especially in the automatic translation and natural language generation tasks [162, 163], have been thoroughly documented in the NLP community. Unsurprisingly, these issues also translate to V&L tasks, and are often further exacerbated by the added requirement of V&L integration. In V&L tasks, language can be used to express similar visual semantic content in different ways, which makes automatic evaluation of models that emit words and sentences particularly challenging. For example, the captions 'A man is walking next to a tree' and 'A guy is taking a stroll by the tree' are nearly identical in meaning, but it can be hard for automatic systems to infer that fact. Several evaluation metrics have been proposed for captioning, including simple n-gram matching systems (e.g., BLEU [51], CIDEr [54] and ROUGE [52]) and human consensus-based measures [54]. Most of these metrics have limitations [57, 164], with n-gram based metrics suffering immensely for sentences that are phrased differently but have identical meaning or use synonyms [164]. Alarmingly, evaluation metrics often rank machine-generated captions as being better than human captions but fail when human subjectivity is taken into account [57, 164]. Even humans find it hard to agree on what a 'good' caption entails [54]. Automatic evaluation of captioning is further complicated because it is not clear what is expected from the captioning system. A given image can have many valid captions ranging from descriptions of specific objects in an image, to an overall description of the entire image. However, due to natural regularities and photographer bias, generic captions can apply to a large number of images, thereby gaining high evaluation scores without demonstrating visual understanding [58].

Evaluation issues are lessened in VQA and RER where the output is better defined; however, it is not completely resolved. If performance for VQA is measured using exact answer matches, then even small variations will be harshly punished, e.g., if a model predicts 'bird' instead of 'eagle', then the algorithm is punished as harshly as if it were to predict 'table.' Several solutions have been proposed, but they have their own limitations, e.g., Wu-Palmer Similarity (WUPS), a word similarity metric, cannot be used with sentences and phrases. Alternately, consensus based metrics have been explored [12, 78], where multiple annotations are collected for each input, with the intention of capturing common variations of the ground truth answer. However, this paradigm can make many questions *unanswerable* due to low human consensus [17, 30]. Multiple-choice evaluation has been proposed by several benchmarks [12, 80]. While this simplifies evaluation, it takes away a lot of the open-world difficulty from the task and can lead to inflated performance via smart guessing [112].

Dataset biases introduce further complications for evaluation metrics. Inadequate met-

**RER output:** The same bounding box is output for each of these RER inputs: gray truck on the right; blue truck on the right; blue; gray; truck; right

**Image Caption:** Same caption applies to all above images: A city street intersection with a bus next to buildings and people walking on sidewalk.

Figure 7.2: The apparent versus true complexity of V&L tasks. In RER (left), omitting a large amount of text has no effect on the output of the system [165]. Similarly, a seemingly detailed caption (right) can apply to a large number of images from the dataset making it easy to 'guess' based on shallow correlations. While it appears as though the captioning system can identify objects ('bus', 'building', 'people'), spatial relationships ('next to', 'on'), and activities ('walking'). However, it is entirely possible for the captioning system to have 'guessed' the caption by detection of one of the objects in the caption, e.g., a 'bus' or even *a common latent* object such as 'traffic light'.

rics can conflate the issues of bias when the statistical distributions of the training and test sets are not taken into account, artificially inflating performance. Metrics normalized to account for the distribution of training data [17] and diagnostic datasets that artificially perturb the distribution of train and test data [16] have been proposed to remedy this. Furthermore, open-ended V&L language tasks can *potentially* test a variety of skills, ranging from relatively easy sub-tasks (detection of large, well-defined objects), to fairly difficult sub-tasks (fine-grained attribute detection, spatial and compositional reasoning, counting, etc.). However, these tasks are not evenly distributed. Placing all skill types on the same footing can inflate system scores and hide how fragile these systems are. Dividing the dataset into underlying tasks can help [17], but the best way to make such a division is not clearly defined.

### 7.2.3 Are V&L systems 'horses?'

Sturm defines a 'horse' as **'a system that appears as if it is solving a particular problem when it actually is not'** [166]. Of course, the 'horse' here refers to the infamous horse named Clever Hans, thought to be capable of arithmetic and abstract thought but was in reality exploiting the micro-signals provided by its handler and audience. Similar

issues are prevalent in both computer vision and NLP, where it is possible for models to arrive at a correct answer by simply exploiting spurious statistical *cues* rather than through robust understanding of the underlying problem. This results in algorithms that achieve higher accuracy but are brittle when subjected to *stress-tests*. For example, in computer vision, CNNs trained on the Imagenet are shown to be biased towards textures rather than the shape resulting in poor generalization to distortions and sub-optimal object detection performance [167]. In NLP, these issues are even more prevalent. [168] shows that it is possible to *guess* the correct answer in a conversational question-answering task by exploiting cues in the prior conversation for up-to 84% of the time. Similarly, in natural language inference (NLI), where the task is to determine whether a hypothesis is *neutral*, an *entailment*, or a *contradiction* to the given premise, a hypothesis-only baseline (which has not seen the premise) significantly outperforms majority-class baseline [169]. This shows that exploiting statistical *cues* contributes to inflated performance. [170] shows similar effects of spurious correlations in argument reasoning comprehension. As V&L research inherits from these research, similar issues are highly prevalent in V&L research. In this section, we review several of these issues and highlight existing studies that scrutinize the true capabilities of existing V&L systems to assess whether they are 'horses'.

**Superficial correlations and true vs. apparent difficulty**

Due to superficial correlations, the difficulty of V&L datasets may be much lower than the true difficulty of comprehensively solving the task (see Fig. 7.2). We outline some of the key studies and their findings that suggest V&L algorithms are relying on superficial correlations that enable them to achieve high performance in common situations but make them vulnerable when tested under different, but not especially unusual, conditions.

**VQA:** Image-blind algorithms that only see questions often perform surprisingly well [30, 91], sometimes even surpassing the algorithms having access to both [30]. Algorithms also often provide inconsistent answers due to irrelevant changes in phrasing [13, 171], signifying a lack of question comprehension. When a VQA dataset is divided into different question-types, algorithms performed well only on easier tasks that CNNs alone excel at, e.g., detecting whether an object is present, but they performed poorly for complex questions that require bi-modal reasoning [17]. This discrepancy in accuracy is not clearly conveyed when simpler accuracy metrics are used. In a multi-faceted study, [125] showed several quirks of VQA, including how VQA algorithms converge to an answer without even processing one half of the question and show an inclination to fixate on the same answer when the same question is repeated for a different image. Similarly, [80] showed that VQA algorithm performance deteriorates when tested on pairs of images that have opposite answers. As shown in Fig. 7.1, VQA systems can actually amplify bias.

**Image captioning:** In image captioning, simply predicting the caption of the training image with the most similar visual features yields relatively high scores using automatic

evaluation metrics [58]. Captioning algorithms exploit multi-modal distributional similarity [27], and generate captions similar to images in the training set, rather than learning concrete representations of objects and their properties.

**Embodied QA and visual dialog:** EmbodiedQA ostensibly requires navigation, visual information collection, and reasoning, but [172] showed that vision blind algorithms perform competitively. Similarly, visual dialog *should* require understanding both visual content and dialog history [173], but an extremely simple method produces near state-of-the-art performance for visual dialog, despite ignoring both visual and dialog information [173].

**Scene graph parsing:** Predicting scene graphs requires understanding object properties and their relationships to each other. However, [159] showed that objects alone are highly indicative of their relationship labels. They further demonstrated that for a given object pair, simply guessing the most common relation for those objects in the training set yields improved results compared to state-of-the-art methods.

**RER:** In a multi-faceted study of RER, [26] demonstrated multiple alarming issues. The first set of experiments involved tampering with the input referring expression to examine if algorithms properly used the text information. Tampering should reduce performance if algorithms make proper use of text to predict the correct answers. However, their results were relatively unaffected when the words were shuffled and nouns/adjectives were removed from the referring expressions. This signifies that it is possible for algorithms to get high scores without explicitly learning to model the objects, attributes and their relationships. The second set of experiments demonstrated that it is possible to predict correct candidate boxes for over 86% of referring expressions, without ever feeding the referring expression to the system. This demonstrates that algorithms can exploit regularities and biases in these datasets to achieve good performance, making these datasets a poor test of the RER task.

Some recent works have attempted to create more challenging datasets that probe the abilities to properly ground vision and language beyond shallow correlations. In FOIL [64], a single noun from a caption is replaced with another, making the caption invalid. Here the algorithm, must determine if the caption has been *FOILed* and then detect the *FOIL* word and replace it with a correct word. Similarly, in NLVR [65], an algorithm is tasked with finding whether a description applies to a pair of images. Both tasks are extremely difficult for modern V&L algorithms with the best performing system on NLVR limited to around 55% (random guess is 50%), well short of the human performance of over 95%. These benchmarks may provide a challenging test bed that can spur the development of next-generation V&L algorithms. However, they remain limited in scope, with FOIL being restricted to noun replacement for a small number of categories (less than 100 categories from the COCO dataset). Hence, it does not test understanding of attributes or relationships between objects. Similarly, NLVR is difficult, but it lacks additional annotations to aid in the measurement of *why* a model fails, or eventually, why

it succeeds.

**Lack of interpretability and confidence**

Human beings can provide explanations, point to evidence, and convey confidence in their predictions. They also have an ability to say 'I do not know' when the information provided is insufficient. However, almost none of the existing V&L algorithms are equipped with these abilities, making the models highly uninterpretable and unreliable.

In VQA, algorithms provide high-confidence answers even when the question is nonsensical for a given image, e.g., 'What color is the horse?' for an image that does not contain a horse can yield 'brown' with a very high confidence. Very limited work has been done in V&L to assess a system's ability to deal with lack of information. While [17] proposed a class of questions called 'absurd' questions to test a system's ability to determine if a question was unanswerable, they were limited in scope to simple detection questions. More complex forms of absurdity are yet to be tested.

Because VQA and captioning do not explicitly require or test for proper grounding or pointing to evidence, the predictions made by these algorithms remain uninterpretable. A commonly practiced remedy is to include visualization of attention maps for attention-based methods, or use post-prediction visualization methods such as Grad-CAM [174]. However, these visualizations shed little light on whether the models have 'attended' to the right image regions. First, most V&L datasets do not contain attention maps that can be compared to the predicted attention maps; therefore, it is difficult to gauge the prediction quality. Second, even if such data were available, it is not clear what image regions the model *should* be looking at. Even for well-defined tasks such as VQA, answers to questions like 'Is it sunny?' can be inferred using multiple image regions. Indeed, inclusion of attention maps does not make a model more predictable for human observers [175], and the attention-based models and humans do not *look* at same image regions [126]. This suggests attention maps are an unreliable means of conveying interpretable predictions.

Several works propose the use of textual explanations to improve interpretability [176, 177]. [177] collected text explanations in conjunction with standard VQA pairs and a model must predict both the correct answer and the explanation. However, learning to predict explanations can suffer from many of the same problems faced by image captioning: evaluation is difficult and there can be multiple valid explanations. Currently, there is no reliable evidence that such explanations actually make the model more interpretable, but there is some evidence of the contrary [175].

Modular and compositional approaches attempt to reveal greater insight by incorporating interpretability directly into the design of the network [84, 102, 122]. However, these algorithms are primarily tested on simpler, synthetically constructed datasets that lack the diversity of natural images and language. The exceptions that are tested on natu-

ral images rely on hand-crafted semantic parsers to pre-process the questions [122], which often over-simplify the complexity of the questions [13].

**Lack of compositional concept learning**

It is hard to verify that a model has understood concepts. One method to do this is to use it in a novel setting or in a previously unseen combination. For example, most humans would not have a problem recognizing a purple colored dog, even if they have never seen one before, given that they are familiar with the concepts of purple and dog. Measuring such compositional reasoning could be crucial in determining whether a V&L system is a 'horse.' This idea has received little attention, with few works devoted to it [15, 83]. Ideally, an algorithm should not show any decline in performance for novel concept combinations. However, even for CLEVR, which is composed of basic geometric shapes and colors, most algorithms show a large drop in performance for novel shape-color combinations [83]. For natural images, the drop in performance is even higher [15].

## 7.3 Conclusion

While V&L work initially seemed incredibly difficult, rapid progress on benchmarks made it appear as if systems would soon rival humans. In this chapter, we argued that much of this progress may be misleading due to dataset bias, superficial correlations and flaws in standard evaluation metrics. While this should serve as a cautionary tale for future research in other areas, we believe V&L research does have a bright future. The vast majority of current V&L research is on creating new algorithms, however, we argue that constructing good datasets and evaluation techniques is just as critical, if not more so, for progress to continue. We discuss potential future research directions to address these shortcomings in the next chapter.

# Chapter 8

# Conclusion and Future Work

In this dissertation, we outlined the progress towards language-grounded visual learning. In a nutshell, we explored three major facets towards that goal.

1. Critical analysis of datasets, evaluation metrics and algorithms and their successes in visual and linguistic understanding. Major issues we discussed include analysis of bias in distribution of questions and answers in VQA dataset (Chapter 4), lack of robustness and proper grounding (Chapter 2). Finally, we also discussed various challenges in broader vision and language (V&L) research (Chapter 7).

2. Creation of novel tasks and evaluation metrics to address several shortcomings in existing vision and language problems. Major directions along these lines include creation of task-directed visual understanding (Chapter 4), exploration of areas such as OCR integration and handling out-of-vocabulary tokens that are traditionally ignored by VQA datasets (Chapter 5).

3. Development of novel vision and language algorithms. We explored the use of predicted answer-type to guide discovery of visual features (Chapter 3), VQA algorithms capable of optical character recognition (OCR) and parsing out-of-vocabulary questions and answers (Chapter 5), and algorithms for learning efficient bi-modal embedding for answering questions about data visualization (Chapter 6).

While we made a lot of progress in this dissertation, there are several immediate and long-term research directions that can further advance the state-of-the-art. In the next sections, we will discuss future research directions that can advance the state of AI for vision and language problems. Building from earlier chapters, we will specifically discuss two key areas: 1) The future of CQA research in light of strong results from our PReFIL algorithm (Chapter 5) and 2) The future work needed to address the challenges in V&L research that we discussed in chapter 7.

## 8.1 Future of CQA Research

The strong results achieved by the PReFIL algorithm in Chapter 6 suggests that the community is ready for new challenges in CQA and related areas. There are three specific avenues that are particularly pertinent.

- **Charts in the wild:** The charts in FigureQA and DVQA are methodologically generated, but human-generated charts in real-world business and scientific documents can contain variations that these datasets omit. Additional text in the chart or human annotations would likely cause the dynamic encoding method used by PReFIL to fail. Next generation datasets should contain charts extracted from real-world documents.
- **Human generated questions:** The questions in both FigureQA and DVQA were created with templates, which do not capture all the nuances of natural language. Deploying a chart question answering system will require it to handle human-generated queries. Studies on the synthetically generated CLEVR dataset have demonstrated that algorithms experience a large drop in performance when natural language questions are asked to a model trained only on CLEVR [84]. Future CQA datasets should include human-generated question-answer pairs.
- **Document-level CQA:** FigureQA and DVQA have well-defined image regions and all information needed to answer a question is contained in that image. To understand charts in documents, information in the rest of the document may be necessary to answer questions about the chart. Beyond typical CQA algorithm abilities, this requires document question answering [178], page segmentation [179], and more. Creation of such a dataset would greatly increase the challenge for future algorithms and better match real-world usage.

## 8.2 Addressing Shortcomings in Vision and Language Research

In the preceding chapter, we complied a wide range of shortcomings and challenges faced by modern V&L research based on the datasets and evaluation of tasks. One of the major issues stems from the difficulty in evaluating if an algorithm is actually solving the task, which is confounded by hidden perverse incentives in modern datasets that cause algorithms to exploit unwanted correlations. Lamentably, most proposed tasks do not have built-in safeguards against this or even an ability to measure it. Many *post-hoc* studies have shed light on this problem. However, they are often limited in scope, require collecting additional data [64], or the modification of 'standard' datasets [15, 16, 30]. We outline prospects for future research in V&L, with an emphasis on discussing the characteristics of future V&L tasks and evaluation suites that are better aligned with the goals of a visual

Table 8.1: A summary of challenges and potential solutions for V&L problems.

| Shortcomings/Challenges | Potential Solutions |
|---|---|
| Evaluation metrics are a poor measure for competence of algorithms due to dataset bias. | • Use metrics that account for dataset biases.<br>• Carefully measure and report performance on individual abilities. |
| It is hard to tell if algorithms are 'right for the right reasons.' They can perform well on benchmarks without actually solving the problem. | • Test the algorithms by withholding varying degrees of task-critical information from them to measure if they understand concepts.<br>• Measure task understanding by asking the model to do the same task in dissimilar contexts and with alternative phrasing.<br>• Develop defense mechanisms against 'accidentally' reaching the correct solutions. |
| Trained systems are fragile and easily break when humans use them. | • Incorporate prediction confidence into evaluation.<br>• Allow systems to output 'I dont know.' |
| V&L Systems are one-trick-ponies, rarely able to generalize to more than one task. | • Create a V&L decathlon that tests numerous V&L tasks. Assess positive transfer among tasks. |

Turing test. Table 8.1 presents a short summary of challenges and potential solutions in V&L research.

### 8.2.1 New V&L tasks that measure core abilities

Existing V&L evaluation schemes for natural datasets ignore bias, making it possible for algorithms to excel on standard benchmarks without demonstrating proper understanding of underlying visual, linguistic or reasoning challenges. We argue that a carefully designed suite of tasks could be used to address this obstacle. We propose some possible approaches to improve evaluation by tightly controlling the evaluation of core abilities and ensuring that evaluation compensates for bias.

CLEVR [83] enables measurement of compositional reasoning, but the questions and scenes have limited complexity. We argue that a CLEVR-like dataset for natural images could be created by composing scenes of natural objects (see Fig. 8.1). This could be used to test higher-levels of visual knowledge, which is not possible in synthetic environments. This approach could be used to examine reasoning and bias-resistance by placing objects

Figure 8.1: *Posters* dataset can help test bias. In this example, both contextual and gender bias are tested by placing out-of-context poster-cut-outs. Snowboarding is generally correlated with gender 'male' and context 'snow' [28].

in unknown combinations and then asking questions with long reasoning chains, novel concept compositions, and distinct train/test distributions.Current benchmarks cannot reliably ascertain whether an algorithm has learned to represent objects and their attributes properly, and it is often easy to produce a correct response by 'guessing' prominent objects in the scene [26]. To examine whether an algorithm demonstrates concept understanding, we envision a dataset containing simple queries, where given a set of objects and/or attributes as queries, the algorithm needs to highlight *all* objects that satisfy *all* of the conditions in the set, e.g., for *query*=$\{red\}$, the algorithm must detect all red objects, and for $\{red,car\}$, it must detect all red cars. However, all queries would have *distractors* in the scene, e.g., $\{red,\ car\}$ is only used when the scene also contains 1) cars that are non-red, 2) objects other than cars or 3) other non-red objects. By abandoning the complexity of natural language, this dataset allows for the creation of queries that are hard to 'guess' without learning proper object and attribute representations. Since the chance of a random guess being successful is inversely proportional to the number of *distractors*, the scoring can also be made proportional to *additional* information over a random guess. While this dataset greatly simplifies the language requirement, it would provide better measurement of elementary language grounded visual concept learning.

Similarly, the core abilities needed for language understanding can be tested using linguistic variations applied to the same visual input. Keeping the visual input unchanged can allow natural language semantic understanding to be better studied. Recent works have done this by rephrasing queries [180]. To some extent, this can be done automatically by merging/negating existing queries, replacing words with synonyms, and introducing

distractors.

We hope that carefully designed test suites that measure core abilities of V&L systems in a controlled manner will be developed. This serves as a necessary adjunct to more open-ended benchmarks, and would help dispel the 'horse' in V&L.

### 8.2.2 Better evaluation of V&L systems

V&L needs better evaluation metrics for standard benchmarks. Here, we will outline some of the key points future evaluation metrics should account for:

- Evaluation should test individual skills to account for dataset biases [17] and measure performance relative to 'shallow' guessing [13, 16, 26].
- Evaluation should include built-in tests for 'bad' or 'absurd' queries [17, 26].
- Test sets should contain a large number of compositionally novel instances that can be inferred from training but not directly matched to a training instance [58, 83].
- Evaluation should keep the 'triviality' of the task in mind when assigning score to a task, e.g., if there is only a single cat then 'Is there a black cat sitting between the sofa and the table?' reduces to 'Is there a cat?' for that image [26, 125].
- Robustness to semantically identical queries must be assessed.
- Evaluation should be done on questions with unambiguous answers; if humans strongly disagree, it is likely not a good question for a visual Turing test.

We believe future evaluation should probe algorithms from multiple angles such that a single score is derived from a suite of sub-scores that test different capabilities. The score could be divided into underlying core abilities (e.g., counting, object detection, fine-grained recognition, etc.), and also higher-level functions (e.g., consistency, predictability, compositionality, resistance to bias, etc.)

### 8.2.3 V&L decathlon

Most of the V&L tasks seek to measure language grounded visual understanding. Therefore, it is not unreasonable to expect an algorithm designed for one benchmark to readily transfer to other V&L tasks with only minor modifications. However, most algorithms are tested on single task [30, 91, 165], with very few exceptions [101, 104, 152]. Even within the same task, algorithms are almost never evaluated on multiple datasets to assess different skills, which makes it difficult to study the true capabilities of the algorithms.

To measure holistic progress in V&L research, we believe it is imperative to create a large-scale V&L decathlon benchmark. Work in a similar spirit has recently been proposed as DecaNLP [181], where many constituent NLP tasks are represented in a single

benchmark. In DecaNLP, all constituent tasks are represented as question-answering for an easier input-output mapping. To be effective, a V&L decathlon benchmark should not only contain different sub-tasks and diagnostic information but also entirely different input-output paradigms. We envision models developed for a V&L decathlon to have a central V&L core and multiple input-output nodes that the model selects based on the input. Both training and test splits of the decathlon should consist of many different input-output mappings representing distinct V&L tasks. For example, the same image could have a **VQA question** 'What color is the cat?', a **pointing question** 'What is the color of "that" object?', where "that" is a bounding box pointing to an object, and a **RER** 'Show me the red cat.' Integration of different tasks encourages development of more capable V&L models. Finally, the test set should contain unanswerable queries [17, 26], compositionally novel instances [15, 84], pairs of instances with subtle differences [80], equivalent queries with same ground truth but different phrasings, and many other quirks that allow us to peer deeper into the reliability and true capacity of the models. These instances can then be used to produce a suite of metrics as discussed earlier.

V&L research has the potential to be a visual Turing test for assessing progress in AI, and we believe that future research along the directions that we proposed will foster the creation of V&L systems that are trustworthy and robust.

# Bibliography


[1] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in neural information processing systems*, 2017, pp. 4967–4976.

[2] S. E. Kahou, A. Atkinson, V. Michalski, Á. Kádár, A. Trischler, and Y. Bengio, "Figureqa: An annotated figure dataset for visual reasoning," *CoRR*, vol. abs/1710.07300, 2017.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[10] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," in *International Conference on Learning Representations (ICLR)*, 2015.

[11] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4584–4593.

[12] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[13] K. Kafle and C. Kanan, "Visual question answering: Datasets, algorithms, and future challenges," *Computer Vision and Image Understanding*, 2017.

[14] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] A. Agrawal, A. Kembhavi, D. Batra, and D. Parikh, "C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset," *arXiv preprint arXiv:1704.08243*, 2017.

[16] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Dont just assume; look and answer: Overcoming priors for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4971–4980.

[17] K. Kafle and C. Kanan, "An analysis of visual question answering algorithms," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 1983–1991.

[18] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.

[19] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "An ontology approach to object-based image retrieval," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2. IEEE, 2003, pp. II–511.

[20] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678.

[21] M. Yatskar, L. Zettlemoyer, and A. Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5534–5542.

[22] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," in *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2017.

[23] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

[24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision ()*, 2017.

[25] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[26] V. Cirik, L.-P. Morency, and T. Berg-Kirkpatrick, "Visual referring expression recognition: What do systems actually learn?" *arXiv preprint arXiv:1805.11818*, 2018.

[27] P. Madhyastha, J. Wang, and L. Specia, "End-to-end image captioning exploits multimodal distributional similarity," in *29th British Machine Vision Conference*. BMVA, 2018.

[28] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women also snowboard: Overcoming bias in captioning models," in *ECCV*, 2018.

[29] K. Kafle, R. Shrestha, and C. Kanan, "Challenges and prospects in vision and language research," *Frontiers in Artificial Intelligence*, vol. 2, p. 28, 2019.

[30] K. Kafle and C. Kanan, "Answer-type prediction for visual question answering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] K. Kafle, B. Price, S. Cohen, and C. Kanan, "Dvqa: Understanding data visualizations via question answering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5648–5656.

[32] K. Kafle, R. Shrestha, and C. Kanan, "Answering questions about data visualizations using efficient bimodal fusion," in *WACV*, 2020.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *International Journal of Computer Vision*, 2015.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[35] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2145–2158.

[36] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[37] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 249–266, 2019.

[38] M. Saeidi, M. Bartolo, P. Lewis, S. Singh, T. Rocktäschel, M. Sheldon, G. Bouchard, and S. Riedel, "Interpretation of natural language rules in conversational machine reading," in *EMNLP*, 2018.

[39] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *SIGKDD Explorations*, vol. 19, pp. 25–35, 2017.

[40] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, 2015.

[41] M. Malinowski and M. Fritz, "Towards a visual turing challenge," *arXiv preprint arXiv:1410.8027*, 2014.

[42] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[43] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[44] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[45] N. Silberman, D. Sontag, and R. Fergus, "Instance segmentation of indoor scenes using a coverage loss," in *European Conference on Computer Vision (ECCV)*, 2014.

[46] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun, "Monocular object instance segmentation and depth ordering with CNNs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[47] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation with deep densely connected MRFs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[49] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[50] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning (ICML)*, 2015.

[51] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[52] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8. Barcelona, Spain, 2004.

[53] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, vol. 29, 2005, pp. 65–72.

[54] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[55] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluation the role of bleu in machine translation research," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[56] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt *et al.*, "From captions to visual concepts and back," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[57] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, "Automatic description generation from images: A survey of models, datasets, and evaluation measures," *Journal of Artificial Intelligence Research*, vol. 55, pp. 409–442, 2016.

[58] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick, "Exploring nearest neighbor approaches for image captioning," *arXiv preprint arXiv:1505.04467*, 2015.

[59] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4565–4574.

[60] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[61] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 817–834.

[62] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[63] M. Acharya, K. Jariwala, and C. Kanan, "Vqd: Visual query detection in natural scenes," in *Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[64] R. Shekhar, S. Pezzelle, Y. Klimovich, A. Herbelot, M. Nabi, E. Sangineto, and R. Bernardi, ""foil it! find one mismatch between image and language caption"," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 255–265.

[65] A. Suhr, M. Lewis, J. Yeh, and Y. Artzi, "A corpus of natural language for visual reasoning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2017, pp. 217–223.

[66] A. Suhr, S. Zhou, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," *arXiv preprint arXiv:1811.00491*, 2018.

[67] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 5, 2018, p. 14.

[68] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[69] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, "Learning cooperative visual dialog agents with deep reinforcement learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[70] J.-H. Kim, D. Parikh, D. Batra, B.-T. Zhang, and Y. Tian, "Codraw: Visual dialog for collaborative drawing," *CoRR*, vol. abs/1712.05558, 2017.

[71] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[72] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? Dataset and methods for multilingual image question answering," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[73] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[74] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[75] Q. Wu, P. Wang, C. Shen, A. van den Hengel, and A. R. Dick, "Ask me anything: Free-form visual question answering based on knowledge from external sources," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[77] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision (ECCV)*, 2012.

[78] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[79] S. Antol, C. L. Zitnick, and D. Parikh, "Zero-shot learning via visual abstraction," in *European Conference on Computer Vision (ECCV)*, 2014.

[80] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 3.

[81] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[82] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.

[83] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1988–1997.

[84] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, "Inferring and executing programs for visual reasoning," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3008–3017.

[85] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.

[86] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[87] S. Kafle and M. Huenerfauth, "Evaluating the usability of automatically generated captions for people who are deaf or hard of hearing," in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, 2017, pp. 165–174.

[88] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2016.

[89] H. Noh, P. H. Seo, and B. Han, "Image question answering using convolutional neural network with dynamic parameter prediction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[90] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision (ECCV)*, 2016.

[91] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[92] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," in *Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016.

[93] I. Ilievski, S. Yan, and J. Feng, "A focused dynamic attention model for visual question answering," *arXiv preprint arXiv:1604.01485*, 2016.

[94] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International Conference on Machine Learning (ICML)*, 2016.

[95] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang, "Multimodal residual learning for visual qa," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[96] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[97] H. Noh and B. Han, "Training recurrent answering units with joint loss minimization for VQA," *arXiv preprint arXiv:1606.03647*, 2016.

[98] H. Nam, J.-W. Ha, and J. Kim, "Dual attention networks for multimodal reasoning and matching," *arXiv preprint arXiv:1611.00471*, 2016.

[99] J.-H. Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016.

[100] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[101] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[102] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[103] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," in *Advances in Neural Information Processing Systems*, 2018, pp. 1039–1050.

[104] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *NeurIPS*, 2018.

[105] J. Zhang, Q. Wu, C. Shen, J. Zhang, J. Lu, and A. Van Den Hengel, "Goal-oriented visual question generation via intermediate rewards," in *European Conference on Computer Vision*. Springer, 2018, pp. 189–204.

[106] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[107] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8317–8326.

[108] Q. Cao, X. Liang, B. Li, and L. Lin, "Interpretable visual question answering by reasoning on dependency trees," *arXiv preprint arXiv:1809.01810*, 2018.

[109] K. Saito, A. Shin, Y. Ushiku, and T. Harada, "Dualnet: Domain-invariant network for visual question answering," *arXiv preprint arXiv:1606.06108*, 2016.

[110] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proc. IEEE Int. Conf. Comp. Vis*, vol. 3, 2017.

[111] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[112] A. Jabri, A. Joulin, and L. van der Maaten, "Revisiting visual question answering baselines," in *European Conference on Computer Vision (ECCV)*, 2016.

[113] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4223–4232.

[114] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[115] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *International Conference on Learning Representations (ICLR)*, 2015.

[116] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2015.

[117] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.

[118] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[119] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International Conference on Machine Learning (ICML)*, 2016.

[120] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[121] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," in *ICLR*, 2018.

[122] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 804–813.

[123] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer *et al.*, "DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[124] V. Kazemi and A. Elqursh, "Show, ask, attend, and answer: A strong baseline for visual question answering," *arXiv preprint arXiv:1704.03162*, 2017.

[125] A. Agrawal, D. Batra, and D. Parikh, "Analyzing the behavior of visual question answering models," in *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2016.

[126] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" in *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2016.

[127] A. Torralba and A. Efros, "Unbiased look at dataset bias," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[128] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: data mining, inference and prediction," *New York: Springer-Verlag*, vol. 1, no. 8, pp. 371–406, 2001.

[129] X.-Y. Zhang and C.-L. Liu, "Locally smoothed modified quadratic discriminant function," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2013.

[130] R. Raina, Y. Shen, A. Mccallum, and A. Y. Ng, "Classification with hybrid generative/discriminative models," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[131] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[132] A. Ng and A. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[133] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel, "Visual question answering: A survey of methods and datasets," *arXiv preprint arXiv:1607.05910*, 2016.

[134] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 594–611, 2006.

[135] E. Kim and K. F. McCoy, "Multimodal deep learning using images and text for information graphic classification," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, 2018, pp. 143–148.

[136] R. A. Al-Zaidy and C. L. Giles, "Automatic extraction of data from bar charts," in *Proceedings of the 8th International Conference on Knowledge Capture*. ACM, 2015, p. 30.

[137] S. Elzer, S. Carberry, and I. Zukerman, "The automated understanding of simple bar charts," *Artificial Intelligence*, vol. 175, no. 2, pp. 526–555, 2011.

[138] J. S. Kallimani, K. Srinivasa, and R. B. Eswara, "Extraction and interpretation of charts in technical documents," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. IEEE, 2013, pp. 382–387.

[139] J. Poco and J. Heer, "Reverse-engineering visualizations: Recovering visual encodings from chart images," in *Computer Graphics Forum*, vol. 36, no. 3. Wiley Online Library, 2017, pp. 353–363.

[140] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "Revision: Automated classification, analysis and redesign of chart images," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 393–402.

[141] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, "A diagram is worth a dozen images," in *ECCV*, 2016.

[142] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi, "Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension," in *CVPR*, 2017.

[143] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[144] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[145] K. Kafle, M. Yousefhussien, and C. Kanan, "Data augmentation for visual question answering," in *INLG*, 2017.

[146] R. Reddy, R. Ramesh, A. Deshpande, and M. M. Khapra, "A question-answering framework for plots using deep learning," *arXiv preprint arXiv:1806.04655*, 2018.

[147] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[148] G. Huang, S. Liu, L. van der Maaten, and K. Q. Weinberger, "CondenseNet: An efficient densenet using learned group convolutions," in *CVPR*, 2018.

[149] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[150] M. Malinowski and C. Doersch, "The visual QA devil in the details: The impact of early fusion and batch norm on clevr," *arXiv preprint arXiv:1809.04482*, 2018.

[151] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," in *AAAI*, 2018.

[152] R. Shrestha, K. Kafle, and C. Kanan, "Answer them all! toward universal visual question answering models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[153] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *CVPR*, 2017.

[154] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Domain adaptation in computer vision applications*. Springer, 2017, pp. 37–55.

[155] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.

[156] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in neural information processing systems*, 2016.

[157] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *arXiv preprint arXiv:1805.12177*, 2018.

[158] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen, "Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects," *arXiv preprint arXiv:1811.11553*, 2018.

[159] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[160] A. Godil, R. Bostelman, W. P. Shackleford, T. Hong, and M. O. Shneier, "Performance metrics for evaluating object and human detection and tracking systems," 2014.

[161] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[162] H. Shimanaka, T. Kajiwara, and M. Komachi, "Metric for automatic machine translation evaluation based on universal sentence representations," *arXiv preprint arXiv:1805.07469*, 2018.

[163] J. Novikova, O. Dušek, A. C. Curry, and V. Rieser, "Why we need new evaluation metrics for nlg," *arXiv preprint arXiv:1707.06875*, 2017.

[164] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem, "Re-evaluating automatic metrics for image captioning," in *European Chapter of the Association for Computational Linguistics (EACL)*, 2017.

[165] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg, "Mattnet: Modular attention network for referring expression comprehension," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[166] B. L. Sturm, "The "horse" inside: Seeking causes behind the behaviors of music content analysis systems," *Computers in Entertainment (CIE)*, vol. 14, no. 2, 2016.

[167] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[168] A. Sharma, D. Contractor, H. Kumar, and S. Joshi, "Neural conversational qa: Learning to reason v.s. exploiting patterns," *ArXiv*, vol. abs/1909.03759, 2019.

[169] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. V. Durme, "Hypothesis only baselines in natural language inference," in **SEM@NAACL-HLT*, 2018.

[170] T. Niven and H.-Y. Kao, "Probing neural network comprehension of natural language arguments," *arXiv preprint arXiv:1907.07355*, 2019.

[171] A. Ray, G. T. Burachas, K. Sikka, A. Roy, A. Ziskind, Y. Yao, and A. Divakaran, "Make up your mind: Towards consistent answer predictions in vqa models," in *European Conference on Computer Vision (ECCV), Workshops*, 2018.

[172] A. Anand, E. Belilovsky, K. Kastner, H. Larochelle, and A. Courville, "Blindfold baselines for embodied qa," in *Advances in Neural Information Processing Systems Workshops (NIPS)*, 2018.

[173] D. Massiceti, P. K. Dokania, N. Siddharth, and P. H. Torr, "Visual dialogue without vision or dialogue," *arXiv preprint arXiv:1812.06417*, 2018.

[174] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization." in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[175] A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh, "Do explanations make vqa models more predictable to a human?" *arXiv preprint arXiv:1810.12366*, 2018.

[176] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–19.

[177] Q. Li, Q. Tao, S. Joty, J. Cai, and J. Luo, "Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions," *arXiv preprint arXiv:1803.07464*, 2018.

[178] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *ACL*, 2018.

[179] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *ICDAR*, 2017.

[180] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, "Cycle-consistency for robust visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6649–6658.

[181] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

# Appendix A

# Additional Details in TDIUC

## A.1 Additional Details About TDIUC

In this section, we will provide additional details about the TDIUC dataset creation and additional statistics.

### A.1.1 Questions using Visual Genome Annotations

As mentioned in the main text, Visual Genome's annotations are both non-exhaustive and duplicated. This makes using them to automatically make question-answer (QA) pairs difficult. Due to these issues, we only used them to make two types of questions: Color Attributes and Positional Reasoning. Moreover, a number of restrictions needed to be placed, which are outlined below.

For making Color Attribute questions, we make use of the attributes metadata in the Visual Genome annotations to populate the template 'What color is the `<object>`?' However, Visual Genome metadata can contain several color attributes for the same object as well as different names for the same object. Since the annotators type the name of the object manually rather than choosing from a predetermined set of objects, the same object can be referred by different names, e.g., 'xbox controller,' 'game controller,' 'joystick,' and 'controller' can all refer to same object in an image. The object name is sometimes also accompanied by its color, e.g., 'white horse' instead of 'horse' which makes asking the Color Attribute question 'What color is the white horse?' pointless. One potential solution is to use the wordnet 'synset' which accompanies every object annotation in the Visual Genome annotations. Synsets are used to group different variations of the common objects names under a single noun from wordnet. However, we found that the synset matching was erroneous in numerous instances, where the object category was misrepresented by the given synset. For example, A 'controller' is matched with synset 'accountant' even when the 'controller' is referring to a game controller. Similarly, a 'cd'

Table A.1: The number of questions produced via each source.

| | **Questions** | **Images** | **Unique Answers** |
|---|---|---|---|
| Imported (VQA) | 49,990 | 43,636 | 812 |
| Imported (Genome) | 310,225 | 89,039 | 1,446 |
| Generated (COCO) | 1,286,624 | 122,218 | 108 |
| Generated (Genome) | 6,391 | 5,988 | 675 |
| Manual | 937 | 740 | 218 |
| Grand Total | 1,654,167 | 167,437 | 1,618 |

is matched with synset of 'cadmium.' To avoid these problems we made a set of stringent requirements before making questions:

1. The chosen object should only have a single attribute that belongs to a set of commonly used colors.

2. The chosen object name or synset must be one of the 91 common objects in the MS-COCO annotations.

3. There must be only one instance of the chosen object.

Using these criteria, we found that we could safely ask the question of the form 'What color is the `<object>`?'.

Similarly, for making Positional Reasoning questions, we used the relationships metadata in the Visual Genome annotations. The relationships metadata connects two objects by a relationship phrase. Many of these relationships describe the positions of the two objects, e.g., A is 'on right' of B, where 'on right' is one of the example relationship clause from Visual Genome, with the object A as the subject and the object B as the object. This can be used to generate Positional Reasoning questions. Again, we take several measures to avoid ambiguity. First, we only use objects that appear once in the image because 'What is to the left of A' can be ambiguous if there are two instances of the object A. However, since visual genome annotations are non-exhaustive, there may still (rarely) be more than one instance of object A that was not annotated. To disambiguate such cases, we use the attributes metadata to further specify the object wherever possible, e.g., instead of asking 'What is to the right of the bus?', we ask 'What is to the right of the green bus?'

Due to a these stringent criteria, we could only create a small number of questions using Visual Genome annotations compared to other sources. The number of questions produced via each source is shown in Table A.1.

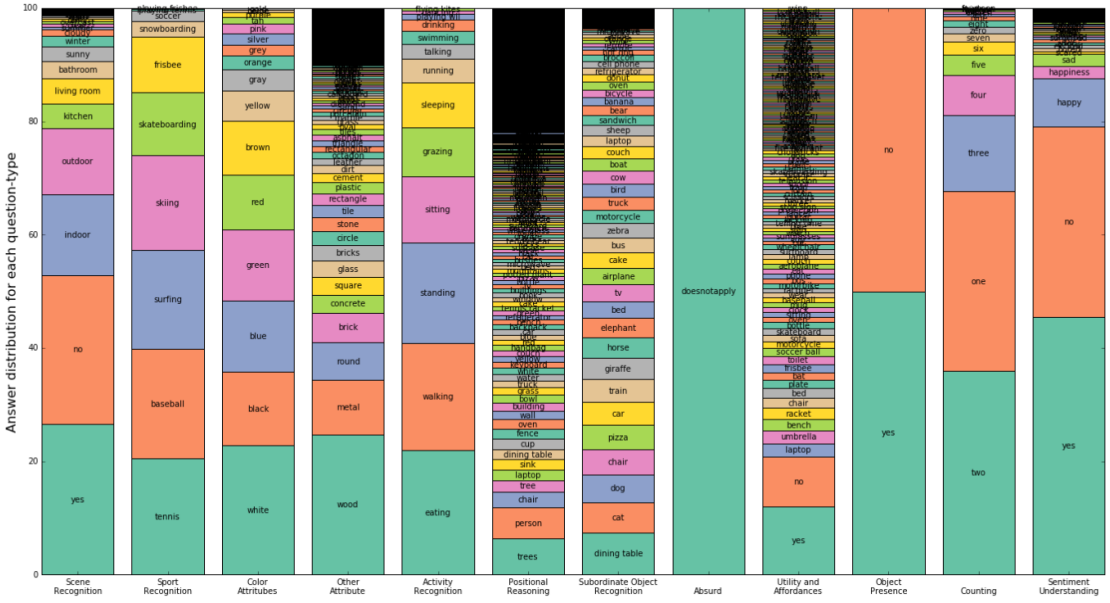



Figure A.1: Answer distributions for the answers for each of the question-types. This shows the relative frequency of each unique answer within a question-type, so for some question-types, e.g., counting, even slim bars contain a fairly large number of instances with that answer. Similarly, for less populated question-types such as utility and affordances, even large bars represents only a small number of training examples.

### A.1.2 Answer Distribution

Figure A.1 shows the answer distribution for the different question-types. We can see that some categories, such as counting, scene recognition and sentiment understanding, have a very large share of questions represented by only a few top answers. In such cases, the performance of a VQA algorithm can be inflated unless the evaluation metric compensates for this bias. In other cases, such as positional reasoning and object utility and affordances, the answers are much more varied, with top-50 answers covering less than 60% of all answers.

We have completely balanced answer distribution for object presence questions, where exactly 50% of questions being answered 'yes' and the remaining 50% of the questions are answered 'no'. For other categories, we have tried to design our question generation algorithms so that a single answer does not have a significant majority within a question type. For example, while scene understanding has top-4 answers covering over 85% of all the questions, there are roughly as many 'no' questions (most common answer) as there are 'yes' questions (second most-common answer). Similar distributions can be seen for counting, where 'two' (most-common answer) is repeated almost as many times as 'one' (second most-common answer). By having at least the top-2 answers split almost equally,

we remove the incentive for an algorithm to perform well using simple mode guessing, even when using the simple accuracy metric.

### A.1.3 Train and Test Split

In chapter 4, we mentioned that we split the entire collection into 70% train and 30% test/validation. To do this, we not only need to have a roughly equal distribution of question types and answers, but also need to make sure that the multiple questions for same image do not end up in two different splits, i.e., the same image cannot occur in both the train and the test partitions. So, we took following measures to split the questions into train-test splits. First, we split all the images into three separate clusters.

1. Manually uploaded images, which includes all the images manually uploaded by our volunteer annotators.
2. Images from the COCO dataset, including all the images for questions generated from COCO annotations and those imported from COCO-VQA dataset. In addition, a large number of Visual Genome questions also refer to COCO images. So, some questions that are generated and imported from Visual Genome are also included in this cluster.
3. Images exclusively in the Visual Genome dataset, which includes images for a part of the questions imported from Visual Genome and those generated using that dataset.

We follow simple rules to split each of these clusters of images into either belonging to the train or test splits.

1. All the questions belonging to images coming from the 'train2014' split of COCO images are assigned to the train split and all the questions belonging to images from the 'val2014' split are assigned to test split.
2. For manual and Visual Genome images, we randomly split 70% of images to train and rest to test.

## A.2 Additional Experimental Results

In this section, we present additional experimental results for chapter 4. First, the detailed normalized scores for each of the question-types is presented in Table 4.3. To compute these scores, the accuracy for each unique answer is calculated separately within a question-type and averaged. Second, we present the results from the experiment in section 4.7.3 in table A.3 (Unnormalized) and tableA.4 (Normalized). The results are evaluated on TDIUC-Tail, which is a subset of TDIUC that only consists of questions that have

Table A.2: Results for all the VQA models. The normalized accuracy for each question-type is shown here. Overall performance is, again, reported using all 5 metrics. Overall (Arithmetic N-MPT) and Overall (Harmonic N-MPT) are averages of the reported sub-scores. Similarly, Arithmetic MPT and Harmonic MPT are averages of sub-scores reported in chapter 4. * denotes training without absurd questions.

| | **YES** | **REP** | **IMG** | **QUES** | **Q+I** | ***Q+I** | **MLP** | **MCB** | ***MCB** | **MCB-A** | **NMN** | **RAU** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scene Recognition | 2.08 | 2.08 | 2.83 | 13.67 | 25.35 | 24.96 | 20.54 | 36.34 | 32.55 | **38.53** | 29.06 | 32.69 |
| Sport Recognition | 0.00 | 9.09 | 12.57 | 11.09 | 51.48 | 60.31 | 60.81 | 75.25 | 73.64 | **75.38** | 63.51 | 73.60 |
| Color Attributes | 0.00 | 6.25 | 1.77 | 20.10 | 25.45 | 30.37 | 30.97 | 36.98 | 37.54 | **49.40** | 33.06 | 46.79 |
| Other Attributes | 0.00 | 0.31 | 1.16 | 6.21 | 6.98 | 9.51 | 2.84 | 13.90 | 15.04 | **15.09** | 7.10 | 12.11 |
| Activity Recognition | 0.00 | 7.69 | 2.88 | 7.59 | 16.09 | 39.35 | 24.95 | 46.57 | 48.27 | **48.47** | 22.79 | 46.65 |
| Positional Reasoning | 0.00 | 0.15 | 0.70 | 4.03 | 6.26 | 8.59 | 2.99 | 9.29 | 9.39 | **10.76** | 6.37 | 9.60 |
| Sub. Object Recognition | 0.00 | 0.47 | 3.16 | 3.72 | 15.91 | 16.97 | 14.85 | 22.07 | 23.05 | **23.22** | 16.83 | 21.67 |
| Absurd | 0.00 | **100.00** | 19.97 | 96.71 | 96.98 | N/A | 95.96 | 83.44 | N/A | 84.82 | 87.51 | 96.08 |
| Utility and Affordances | 1.22 | 1.22 | 1.34 | 9.23 | 16.85 | 21.97 | 6.18 | 24.07 | 23.33 | **26.20** | 19.55 | 21.38 |
| Object Presence | 50.00 | 50.00 | 20.73 | 69.06 | 69.43 | 69.50 | 92.33 | 91.84 | 91.95 | 93.64 | 92.50 | **94.38** |
| Counting | 0.00 | 6.25 | 1.31 | 10.30 | 14.61 | 14.62 | 16.43 | 17.83 | 18.09 | 20.80 | 15.52 | **23.11** |
| Sentiment Understanding | 4.00 | 4.00 | 1.43 | 5.80 | 8.18 | 12.94 | 7.49 | 20.09 | 17.49 | **20.41** | 9.22 | 14.43 |
| Overall (Arithmetic MPT) | 11.10 | 31.11 | 9.49 | 39.31 | 55.25 | 57.03 | 60.87 | 65.75 | 66.07 | **67.90** | 62.59 | 67.81 |
| Overall (Harmonic MPT) | 0.00 | 17.53 | 1.92 | 25.93 | 44.13 | 50.30 | 42.80 | 58.03 | 55.43 | **60.47** | 51.87 | 59.00 |
| Overall (Arithmetic N-MPT) | 4.87 | 15.63 | 5.82 | 21.46 | 29.47 | 28.10 | 31.36 | 39.81 | 35.49 | **42.24** | 34.00 | 41.04 |
| Overall (Harmonic N-MPT) | 0.00 | 0.83 | 1.91 | 8.42 | 14.99 | 18.30 | 9.46 | 24.77 | 23.20 | **27.28** | 16.67 | 23.99 |
| Simple Accuracy | 21.14 | 51.15 | 14.54 | 62.74 | 69.53 | 63.30 | 81.07 | 79.20 | 78.06 | 81.86 | 79.56 | **84.26** |

Table A.3: Results on TDIUC-Tail for MCB model when trained on full TDIUC dataset vs when trained only on TDIUC-Tail. The un-normalized scores for each question-types and five different overall scores are shown here

| | **MCB TDIUC-Full** | **MCB TDIUC-Tail** |
|---|---|---|
| Scene Recognition | 61.64 | 66.59 |
| Sport Recognition | 71.61 | 93.74 |
| Color Attributes | 6.83 | 84.34 |
| Other Attributes | 32.80 | 43.37 |
| Activity Recognition | 51.79 | 74.40 |
| Positional Reasoning | 25.16 | 29.59 |
| Object Recognition | 63.90 | 75.89 |
| Absurd | N/A | N/A |
| Utility and Affordances | 16.67 | 17.59 |
| Object Presence | N/A | N/A |
| Counting | 4.87 | 29.83 |
| Sentiment Understanding | 41.30 | 50.72 |
| Overall (Arithmetic MPT) | 37.66 | 51.61 |
| Overall (Harmonic MPT) | 17.51 | 43.27 |
| Overall (Arithmetic N-MPT) | 19.49 | 34.44 |
| Overall (Harmonic N-MPT) | 11.37 | 22.32 |
| Simple Accuracy | 38.55 | 50.11 |

answers repeated less than 1000 times (uncommon answers). Note that the TDIUC-Tail excludes the absurd and the object presence question-types, as they do not contain any questions with uncommon answers. The algorithms are identical in both Table A.3 and A.4 and are named as follows:

1. **MCB TDIUC-Full** : MCB model trained on whole of the TDIUC dataset and evaluated on TDIUC-Tail.

2. **MCB TDIUC-Tail** : MCB model trained and evaluated on TDIUC-Tail.

Table A.4: Results on TDIUC-Tail for MCB model when trained on full TDIUC dataset vs when trained only on TDIUC-Tail. The normalized scores for each question-types and five different overall scores are shown here

| | **MCB TDIUC-Full** | **MCB TDIUC-Tail** |
|---|---|---|
| Scene Recognition | 24.86 | 29.18 |
| Sport Recognition | 54.82 | 62.74 |
| Color Attributes | 7.03 | 84.40 |
| Other Attributes | 13.04 | 17.01 |
| Activity Recognition | 45.48 | 64.83 |
| Positional Reasoning | 7.46 | 10.99 |
| Object Recognition | 12.55 | 24.20 |
| Absurd | N/A | N/A |
| Utility and Affordances | 12.37 | 14.02 |
| Object Presence | N/A | N/A |
| Counting | 4.87 | 18.96 |
| Sentiment Understanding | 12.45 | 18.08 |
| Overall (Arithmetic MPT) | 37.66 | 51.61 |
| Overall (Harmonic MPT) | 17.51 | 43.27 |
| Overall (Arithmetic N-MPT) | 19.49 | 34.44 |
| Overall (Harmonic N-MPT) | 11.37 | 22.32 |
| Simple Accuracy | 38.55 | 50.11 |

# Appendix B

# Additional Details in DVQA

## B.1 Additional details about the dataset

In this section, we present additional details on the DVQA dataset statistics and how it was generated.

### B.1.1 Data statistics

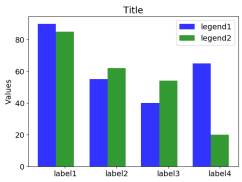
Table B.1 shows the distribution of questions in the DVQA dataset.

### B.1.2 Variations in question templates

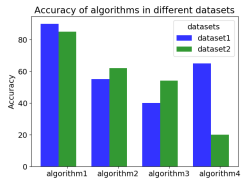
The meaning of different entities in a chart is determined by its title and labels. This allows us to introduce variations in the questions by changing the title of the chart. For example, for a generic title 'Title' and a generic label 'Values', the base-question is: 'What is the value of **L**?'. Depending on the title of the chart, the same question can take following forms:

1. Title: Accuracy of different algorithms, Label: Accuracy ⇒ What is the accuracy of the algorithm **A**?
2. Title: Most preferred objects, Label: Percentage of people ⇒ What percentage of people prefer object **O**?
3. Title: Sales statistics of different items, Label: Units sold ⇒ How many units of the item **I** were sold?

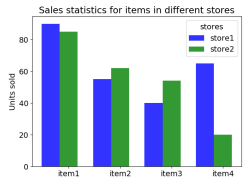
Figure B.1 provides an example on how questions can be varied for the same chart by using a different title and different labels.



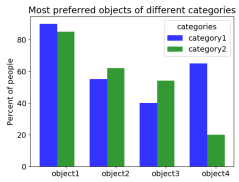

What is the value of *label1* in *legend2*?




What is the accuracy of the algorithm *algorithm1* in the dataset *dataset2*?




How many units of the item *item1* was sold in the store *store2*?




What percentage of people prefer the object *object1* in the category *category2*?

Figure B.1: An example showing that different question can be created by using different title and labels in the same chart.

### B.1.3 Data and visualization generation

In this section, we provide additional details on the heuristics and methods used for generating question-answer pairs.

We aim to design the DVQA dataset such that commonly found visual and data patterns are also more commonly encountered in the DVQA dataset. To achieve this, we downloaded a small sample of bar-charts from Google image search and *loosely* based the distribution of our DVQA dataset on the distribution of downloaded charts. However, some types of chart elements such as logarithmic axes, negative values, etc. that do not occur frequently in the wild are still very important to be studied. To incorporate these in our dataset, we applied such chart elements to a small proportion of the overall dataset. However, we made sure that each of the possible variations was encountered at least 1000 times in the training set.

#### Distribution of visual styles

To incorporate charts with several appearances and styles in our DVQA dataset, we introduced different types of variations in the charts. Some of them as listed below:

1. Variability in the number of bars and/or groups of bars.
2. Single-column vs. multi-column grouped charts.
3. Grouped bars vs. stacked bars. Stacked bars are further divided into two types: 1) Additive stacking, where bars represent individual values, and 2) Fractional stacking, where each bar represents a fraction of the whole.
4. Presence or absence of grid-lines.
5. Hatching and other types of textures.

Table B.1: Statistics on different splits of dataset based on different question types.

| | | **Total Questions** | **Unique Answers** | **Top-2 Answers (in percentage)** |
|---|---|---|---|---|
| Structure | Train | 313,842 | 10 | no: 40.71, yes: 40.71 |
| | Test-Familiar | 78,278 | 10 | no: 41.14, yes: 41.14 |
| | Test-Novel | 78,988 | 10 | no: 41.00, yes: 41.00 |
| Data | Train | 742,896 | 1038 | no: 7.55, yes: 7.55 |
| | Test-Familiar | 185,356 | 1038 | no: 7.44, yes: 7.44 |
| | Test-Novel | 185,452 | 538 | no: 7.51, yes: 7.51 |
| Reasoning | Train | 1,076,391 | 1076 | yes: 8.29, no: 8.26 |
| | Test-Familiar | 268,795 | 1075 | no: 8.31, yes: 8.27 |
| | Test-Novel | 268,788 | 577 | no: 8.28, yes: 8.22 |
| **Overall** | Train | 2,325,316 | 1076 | yes: 11.74, no: 11.73 |
| | Test-Familiar | 580,557 | 1075 | yes: 11.77, no: 11.75 |
| | Test-Novel | 581,321 | 577 | no: 11.80, yes: 11.77 |

6. Text label orientation.
7. A variety of colors, including monochrome styles.
8. Legends placed in a variety of common positions, including legends that are separate from the chart.
9. Bar width and spacing.
10. Varying titles, labels, and legend entries.
11. Vertical vs. horizontal bar orientation.

In the wild, some styles are more common than others. To reflect this in our DVQA dataset, less common styles, *e.g.* hatched bars, are applied to only a small subset of charts. However, every style-choice appears at least a 1000 times in the training set. In overall, 70% of the charts have vertical bars and the remaining charts have horizontal bars. Among multi-column bar-charts, 20% of the linear and normalized percentage bar-charts are presented as stacked bar-charts and the rest are presented as group bar-charts. In legends we have used two styles that are commonly found in the wild: 1) legend below the chart, and 2) legend to the right of the chart. In 40% of the multi-column charts, legends

are positioned outside the bounds of the main chart. Finally, 20% of the charts are hatch-filled with a randomly selected pattern out of six commonly used patterns (stripes, dots, circles, cross-hatch, stars, and grid).

**Distribution of data-types**

Our DVQA dataset contains three major types of data scales.

- **Linear data.** Bar values are chosen from 1 – 10, in an increment of 1. When bars are not stacked, the axis is clipped at 10. When bars are stacked, the maximum value of the axis is automatically set by the height of the tallest stack. For a small number of charts, values are randomly negated or allowed to have missing values (*i.e.* value of zero which appears as a missing bar).
- **Percentage data.** Bar values are randomly chosen from 10–100, in increments of 10. For a fraction of multi-column group bar charts with percentage data, we normalize the data in each group so that the values add up to 100, which is a common style. A small fraction of bars can also have missing or zero value.
- **Exponential data.** Bar values are randomly chosen in the range of 1 - $10^{10}$. The axis is logarithmic.

The majority (70%) of the data in the DVQA dataset is of the linear type (1–10). Among these, 10% of the charts are allowed to have negative. Then, 25% of the data contain percentage scales (10–100), among which half are normalized so that the percentages within each group add up to a 100%. For 10% of both linear and percentage data-type, bars are allowed to have missing (zero) values. The remaining 5% of the data is exponential in nature ranging from $10^0$ – $10^{10}$.

**Ensuring proper size and fit**

Final chart images are drawn such that all of them have the same width and height of $448 \times 448$ pixels. This was done for the ease in processing and to ensure that the images do not need to undergo stretching or aspect ratio change when being processed using an existing CNN architecture. To attain this, we need to ensure that all the elements in the chart fit in the fixed image size. We have taken several steps to ensure a proper fit. By default, the label texts are drawn without rotation *i.e.* horizontally. During this, if any of the texts overlap with each other, we rotate the text by either 45 or 90 degrees. Another issue is when the labels take up too much space leaving too little space for the actual bar-charts, which often makes them illegible. This is usually a problem with styles that contain large texts and/or charts where legend is presented on the side. To mitigate this, we discard the image if the chart-area is less than half of the entire image-area. Similarly, we also discard a chart if we cannot readjust the labels to fit without overlap despite rotating them. Fig. B.2 shows some examples of discarded charts due to poor fit.

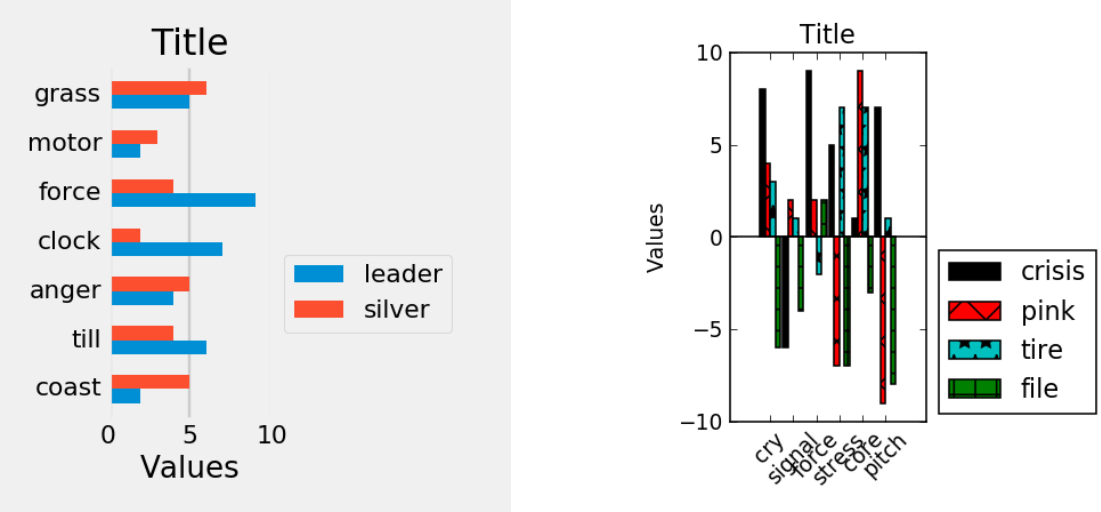



Figure B.2: Examples of discarded visualizations due to the bar-chart being smaller than 50% of the total image area.

### Naming colors

For generating diverse colors, we make use of many of the pre-defined styles that are available with the Matplotlib package and also modify it with several new color schemes. Matplotlib allows us to access the RBG face-color of each drawn bar and legend entries from which we can obtain the color of each of the element drawn in the image. However, to ask questions referring to the color of a bar or a legend entry, we need to be able to name it using natural language (*e.g.* 'What does the `red color` represent?). Moreover, simple names such as 'blue' or 'green' alone may not suffice to distinguish different colors in the chart. So, we employ the following heuristic to obtain a color name for a given RGB value.

1. Start with a dictionary of all 138 colors from the CSS3 X11 named colors. Each of the color is accompanied by its RGB value and its common name. The color names contain names such as darkgreen, skyblue, navy, lavender, chocolate, and other commonly used colors in addition to canonical color names such as 'blue', 'green', or 'red'.

2. Convert all the colors to CIE standard L*a*b* color space which is designed to approximate human perception of the color space.

3. Measure color distance between the L*a*b* color of our chart-element and each of the color in the X11 color dictionary. For distance, we use the CIE 2000 delta

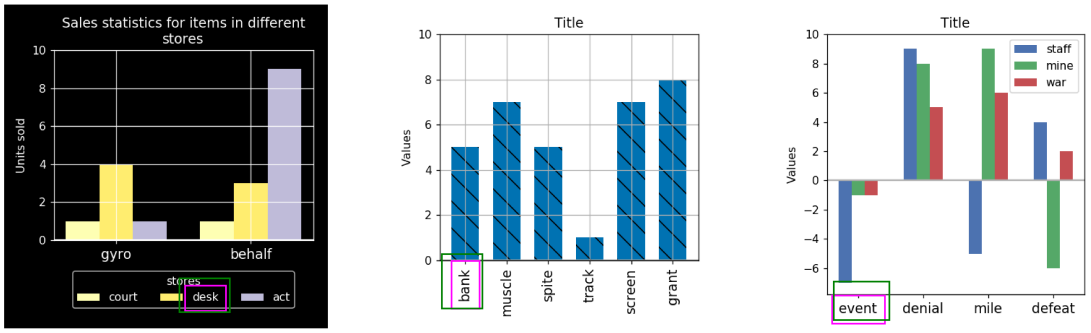



Figure B.3: Some examples showing correctly predicted bounding boxes predicted by our MOM model. Magenta shows the ground truth and green shows the predicted bounding box.

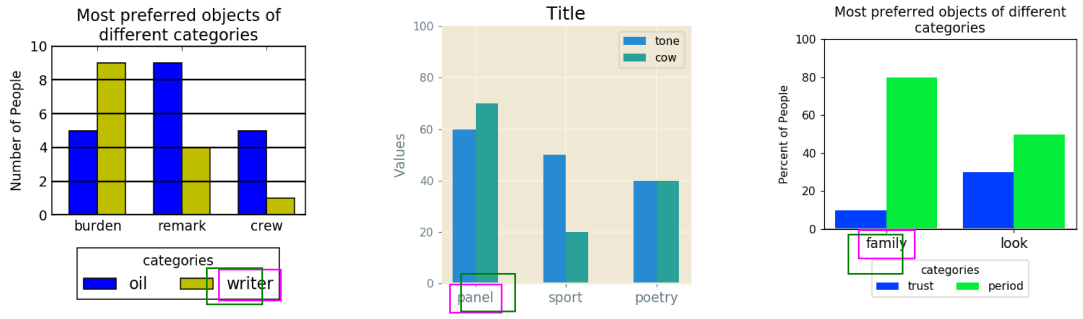



Figure B.4: Some examples showing incorrectly predicted bounding boxes predicted by our MOM model. Often the prediction is off by only a few pixels, but the since the OCR requires total coverage, it results in an erroneous prediction. Magenta shows the ground truth and green shows the predicted bounding box.

E color difference measure which is designed to measure human perceptual differences between colors.

4. Choose the color from the X11 colors which has the lowest delta E value from the color of our chart-element.

## B.2 Analysis of MOM's localization performance

In chapter 5, we observed that many predictions made by MOM were close to the ground truth but not exactly the same. This was also corroborated by taking into account the

Table B.2: Localization performance of MOM in terms of IOU with the ground truth bounding box.

| **IOU with ground truth** | **Percentage of boxes** |
|---|---|
| $\geq 0.2$ | 73.27 |
| $\geq 0.4$ | 56.89 |
| $\geq 0.5$ | 46.06 |
| $\geq 0.6$ | 32.49 |
| $\geq 0.7$ | 18.80 |
| $\geq 0.8$ | 6.93 |
| $\geq 0.9$ | 0.66 |
| $\geq 1.0$ | 0.00 |

Table B.3: Localization performance of MOM in terms of the distance between the center of the predicted and ground truth bounding box.

| **Distance from the ground truth** | **Percentage of boxes** |
|---|---|
| $\leq 1$ pixels | 0.14 |
| $\leq 8$ pixels | 8.48 |
| $\leq 16$ pixels | 25.77 |
| $\leq 32$ pixels | 52.89 |
| $\leq 64$ pixels | 74.21 |

edit-distance between the predicted and ground truth answer strings.

Here we study our hypothesis that this low accuracy is due to poor localization of the predicted bounding boxes. Fig. B.3 shows some results from MOM for Test-Familiar split of the dataset in which the bounding boxes are accurately predicted. This shows that the bounding box prediction network works with texts of different orientations and positions. However, Fig. B.4 shows some examples where boxes do not 'snap' neatly around the text area but are in the right vicinity. Since the OCR subnetwork in MOM operates only on the features extracted from the predicted bounding box, a poor bounding box would also translate to a poor prediction. To quantify this behavior we conduct two separate studies.

First, we measure the intersection over union (IOU) for predicted and ground truth bounding boxes. Table B.2 shows the percentages of boxes that were accurately predicted for various threshold values of IOU.

Next, we measure what percentage of the predicted boxes are within a given distance from the ground truth boxes. The distance is measured as the Euclidean distance between the center x,y co-ordinates for predicted and ground-truth bounding boxes. Result presented in Table B.3 shows that more than half of the predicted boxes are within 32 pixels from the ground truth boxes. Note here that the image dimension is $448 \times 448$ pixels.

The above experiments show that while many of the predicted bounding boxes are 'near' the ground truth boxes, they do not perfectly enclose the text. Therefore, if the predicted bounding boxes are localized better, which could be achieved with additional fine-tuning of the predicted bounding boxes, we can expect a considerable increase in MOM's accuracy on chart-specific answers.

## B.3 Additional examples

In this section, we present additional examples to illustrate the performance of different algorithms for different types of questions. Fig. B.5 shows some example figures with question-answer results for different algorithms and Fig. B.6 shows some interesting failure cases.

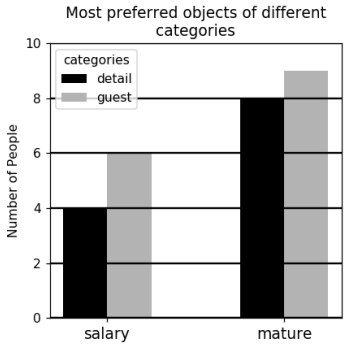
As shown in Fig. B.5, SAN-VQA, MOM, and SANDY all perform with high accuracy across different styles for structure understanding questions. This is unsurprising since all the models use the SAN architecture for answering these questions. However, despite the presence of answer-words in the training set (test-familiar split) SAN is incapable of answering questions with chart-specific answers; it always produces the same answer regardless of the question being asked. In comparison, MOM shows some success in decoding the chart-specific answers. However, as explained earlier in section B.2, the accuracy of MOM for chart-specific answers also depends on the accuracy of the bounding box prediction due to which its predictions were close but not exact for many questions. As discussed in section B.2, although the exact localization of the bounding box was poor, the majority of the predicted bounding boxes were in the vincinity of the ground truth bounding boxes. We believe with additional fine-tuning, *e.g*. regressing for a more exact bounding box based on the features surrounding the initial prediction, could improve the model's performance significantly. Finally, SANDY shows a remarkable success in predicting the chart-specific answers. SANDY's dynamic dictionary converts the task of predicting the answer to predicting the position of the text in the image, making it easier to answer. Once the position is predicted, there are no additional sources of error for SANDY making it less error prone in general.

Similarly, both SAN and MOM are incapable of correctly parsing the questions with chart-specific labels in them. In comparision, SANDY can use the dynamic local dictionary to correctly parse the chart-specific labels showing an improved performance for these questions *e.g*. Fig. B.5c, B.6c, and B.6e.

In Fig. B.6, we study some failure cases to better understand the nature of the errors made by current algorithms. One of the most commonly encountered errors for the algorithms that we tested is the error in predicting exact value of the data. Often, predicting these values involve extracting exact measurement and performing arithmetic operations across different values. The results show that the models are able to perform some mea-

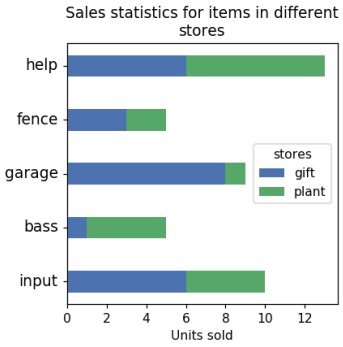
surement; the models predict values that are close to the correct answer, *e.g*. predicting smaller values when the bars have smaller height (Fig. B.6d) and predicting larger values when the bars are tall (Fig. B.6f). In addition, the models are able to make predictions in the accurate data scale *e.g*. For Fig. B.6d, the prediction for the value is in percentage scale (0–100) and for Fig. B.6e, the prediction is in linear scale (0–10).

The next class of the commonly encountered errors is the prediction of chart-specific answers. We have already established that the SAN-VQA model completely fails to answer questions with chart-specific answers, which is demonstrated in all the examples in Fig. B.5 and B.6. Our MOM model also makes errors for several examples as shown in Fig. B.6. The errors occur in decoding the OCR (Fig. B.6a), predicting the right box (Fig. B.6f) or both (Fig. B.6d). While our SANDY model shows vastly increased accuracy for these answers, it can make occasional errors for these questions (Fig. B.6d).



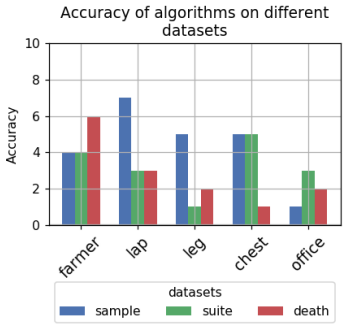

(a)

**Q**: What is the label of the second bar from the left in each group?
SAN: closet ✗ MOM: guest ✓ SANDY: guest ✓
**Q**: Is each bar a single solid color without patterns?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓



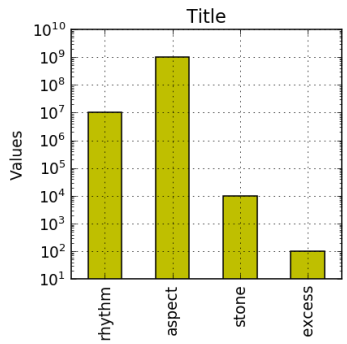

(b)

**Q**: How many items sold less than 6 units in at least one store?
SAN: four ✓ MOM: four ✓ SANDY: four ✓
**Q**: Does the chart contain stacked bars?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓



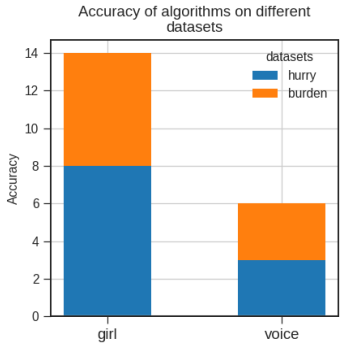

(c)

**Q**: What is the highest accuracy reported in the whole chart?
SAN: 7 ✓ MOM: 7 ✓ SANDY: 7 ✓
**Q**: Is the accuracy of the algorithm leg in the dataset suite smaller than the accuracy of the algorithm chest in the dataset sample?
SAN: no ✗ MOM: no ✗ SANDY: yes ✓



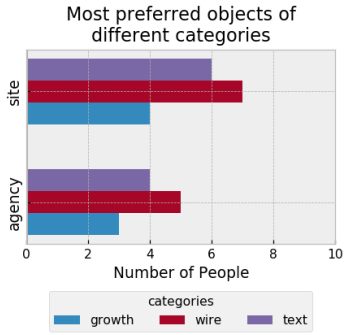

(d)

**Q**: Which bar has the largest value?
SAN: closet ✗ MOM: aspect ✓ SANDY: aspect ✓
**Q**: What is the value of the largest bar?
SAN: $10^9$ ✓ MOM: $10^9$ ✓ SANDY: $10^9$ ✓




(e)

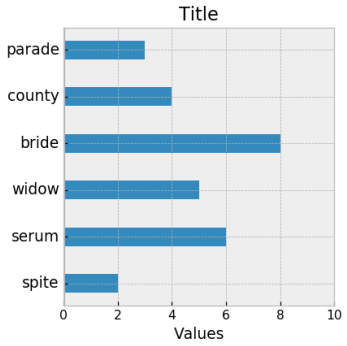
**Q**: How many algorithms have accuracy lower than 3 in at least one dataset?
SAN: zero ✓ MOM: zero ✓ SANDY: zero ✓
**Q**: Which algorithm has highest accuracy for any dataset?
SAN: closet ✗ MOM: girl ✓ SANDY: girl ✓




(f)

**Q**: Which object is preferred by the most number of people summed across all the categories?
SAN: closet ✗ MOM: site ✓ SANDY: site ✓
**Q**: Are the bars horizontal?
SAN: yes ✓ MOM: yes ✓ SANDY: yes ✓

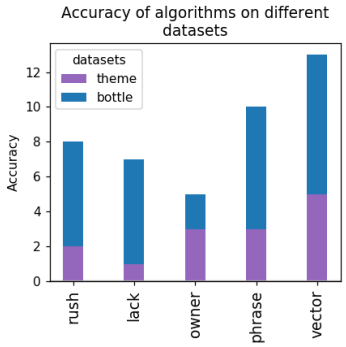
Figure B.5: Some example question-answer pair for different algorithms on the Test-Familiar split of the dataset. The algorithms show success in variety of questions and visualizations. However, the SAN model is utterly incapable of predicting chart-specific answers.




(a)

**Q**: What is the label of the third bar from the bottom?

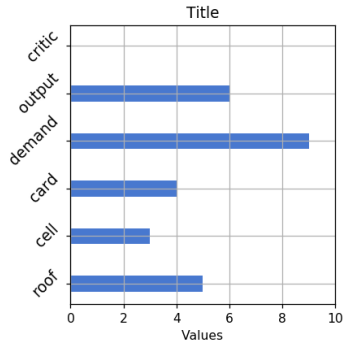
SAN: closet ✗ MOM: whidkw ✗ SANDY: widow ✓




(b)

**Q**: Which algorithm has the largest accuracy summed across all the datasets?

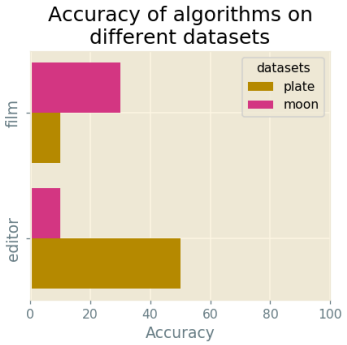
SAN: closet ✗ MOM: lack ✗ SANDY: vector ✓




(c)

**Q**: Is the value of output smaller than demand?

SAN: no ✗ MOM: no ✗ SANDY: yes ✓



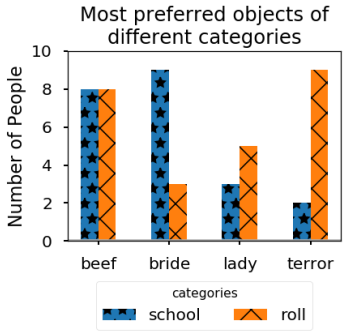

(d)

**Q**: Which algorithm has the smallest accuracy summed across all the datasets?

SAN: closet ✗ MOM: fil ✗ SANDY: editor ✗

**Q**: What is the highest accuracy reported in the whole chart?

SAN: 60 ✗ MOM: 60 ✗ SANDY: 60 ✗



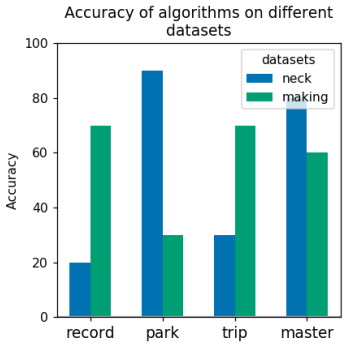

(e)

**Q**: How many total people preferred the object terror across all the categories?

SAN: 10 ✗ MOM: 10 ✗ SANDY: 10 ✗

**Q**: How many people prefer the object terror in the category roll?

SAN: 1 ✗ MOM: 1 ✗ SANDY: 9 ✓




(f)

**Q**: What is the highest accuracy reported in the whole chart?

SAN: 90 ✓ MOM: 90 ✓ SANDY: 80 ✗

**Q**: Which algorithm has the smallest accuracy summed across all the datasets?

SAN: closet ✗ MOM: record ✓ SANDY: park ✗

Figure B.6: Some failure cases for different algorithms on the Test-Familiar split of the dataset.

# Appendix C

# Additional Details for the PReFIL model

Here, we provide additional results and examples for the PReFIL model (Chapter 6).

## C.1 Analysis per FigureQA Question Template

Table C.1 shows results for PReFIL compared to RN [1, 2] and human baselines [2] for different question templates. The results are from a subset of the Test 2 split in FigureQA. As mentioned in chapter 6, Test 2 split consists of chart images where the charts have alternated colors compared to the training set, such that the colors are novel for a given chart-type. Test 2 annotations are not publicly available and the results were obtained by sending model predictions to the authors. As seen in table C.1, PReFIL outperforms RN for all question templates by a large margin and also outperforms human baseline in 12 out of 15 question templates.

## C.2 More Discussion of Example Outputs

We present additional examples for our PReFIL algorithm for both the DVQA [31] (Fig. C.1) and FigureQA (Fig. C.2) datasets. For both datasets, we present examples of correct predictions for a variety of examples (top two rows) and some cases of incorrect predictions (bottom row).

For DVQA, PReFIL with oracle OCR is exceedingly capable, with accuracy of over 96% (see main text for details), but it makes some occasional errors. First, since the dynamic encoding is based on the position of words in the chart, PReFIL may detect the wrong word when the words are in close proximity to each other (Fig. C.1, bottom left). Second, when the chart elements are partially or fully obscured by the legend, PReFIL

Table C.1: Results for PReFIL compared with RN [1,2] and Human baseline [2] compared with each unique question template in FigureQA.

| **Question Template** | **Figure Types** | **RN [1, 2]** | **Human [2]** | **PReFIL** |
|---|---|---|---|---|
| Is X the minimum? | bar, pie | 76.78 | 97.06 | **97.20** |
| Is X the maximum? | bar, pie | 83.47 | 97.18 | **98.07** |
| Is X the low median? | bar, pie | 66.69 | 86.39 | **93.07** |
| Is X the high median? | bar, pie | 66.50 | 86.91 | **93.00** |
| Is X less than Y ? | bar, pie | 80.49 | 96.15 | **98.20** |
| Is X greater than Y ? | bar, pie | 81.00 | 96.15 | **98.07** |
| Does X have the minimum area under the curve? | line | 69.57 | **94.22** | 94.00 |
| Does X have the maximum area under the curve? | line | 78.45 | 95.36 | **96.91** |
| Is X the smoothest? | line | 58.57 | **78.02** | 71.87 |
| Is X the roughest? | line | 56.28 | **79.52** | 74.67 |
| Does X have the lowest value? | line | 69.65 | 90.33 | **92.17** |
| Does X have the highest value? | line | 76.23 | 93.11 | **94.83** |
| Is X less than Y? | line | 67.75 | 90.12 | **92.38** |
| Is X greater than Y? | line | 67.12 | 89.88 | **92.00** |
| Does X intersect Y ? | line | 68.75 | 89.62 | **91.25** |
| Overall | bar,pie,line | 72.18 | 91.21 | **92.79** |

often fails to correctly parse the chart data (Fig. C.1, bottom center). Finally, for some charts, questions involving multiple measurements are also erroneous, especially when the measurements differ only by a small amount (Fig. C.1, bottom right).

For FigureQA, PReFIL again performs well across all categories, surpassing overall human accuracy. PReFIL is capable of answering a wide range of questions across several types of images (Fig. C.2, top 2 rows). However, PReFIL often struggles for question template "Is X the smoothest/roughest?" especially for the dot-line style graphs. The errors are more prominent when the legend obscures or intermingles with the chart elements (Fig. C.2, bottom left). Since the dots are not connected to each other, it is an extremely difficult task even for attentive human observers. Similarly, PReFIL makes occasional mistakes when comparing elements that are very close to each other (Fig. C.2, bottom center and right). However, as seen in Table C.1, PReFIL is more accurate than even human observers for comparing two elements.

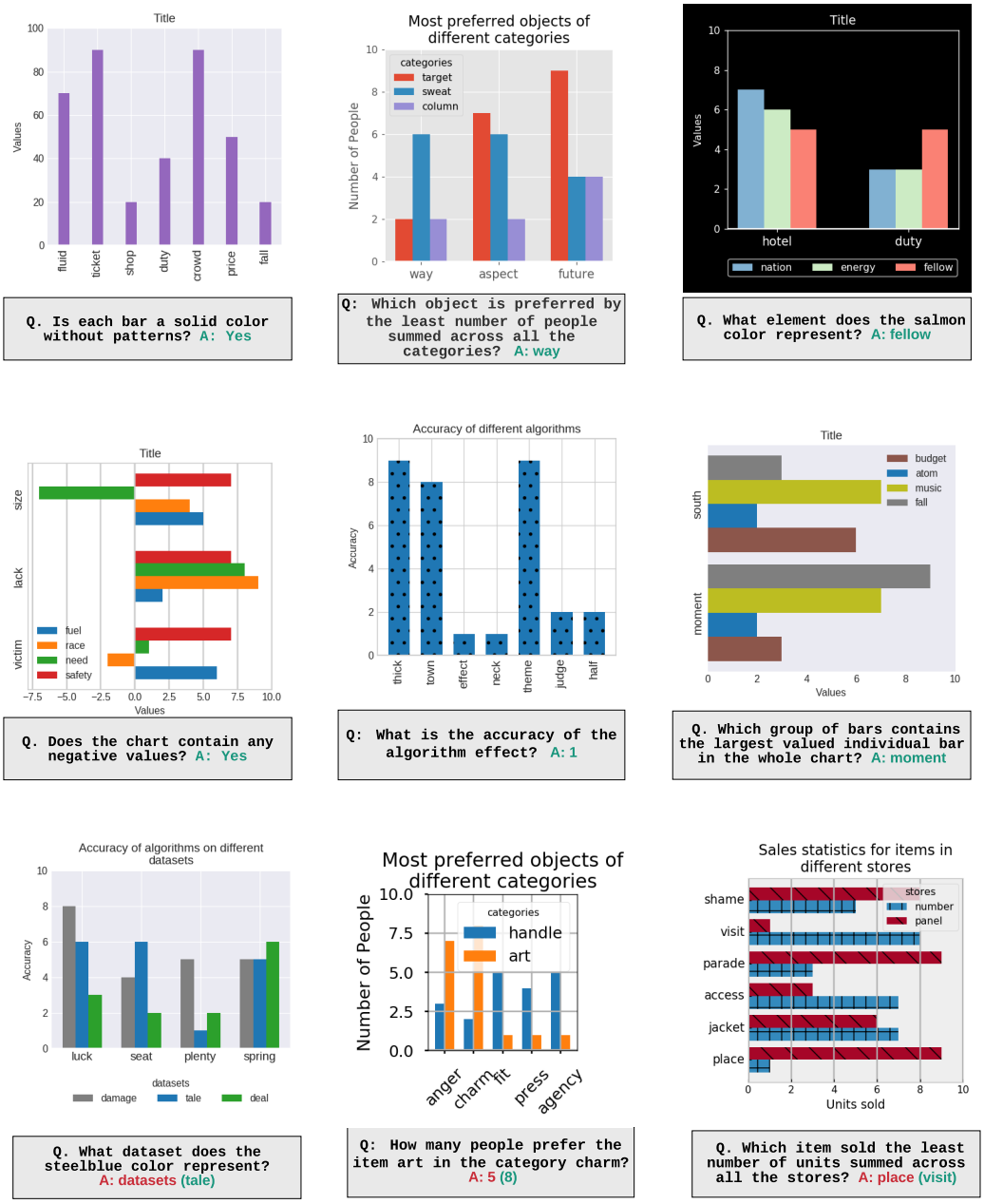



Figure C.1: Some example predictions for PReFIL on the DVQA dataset. Red denotes incorrect predictions. For incorrect predictions, correct answer is shown in parenthesis.


Figure C.2: Some example predictions for PReFIL on the FigureQA dataset. Bottom row shows some incorrect predictions made by PReFIL. Red denotes incorrect predictions. For incorrect predictions, correct answer is shown in parenthesis.